

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-191090

(43)Date of publication of application : 21.07.1998

(51)Int.Cl.

H04N 1/60
 B41J 2/525
 G06T 1/00
 G09G 5/06
 H04N 1/46

(21)Application number : 09-274697

(71)Applicant : SEIKO EPSON CORP

(22)Date of filing : 07.10.1997

(72)Inventor : FUKAZAWA KENJI
 KASAHARA HIROKAZU

(30)Priority

Priority number : 08282726

Priority date : 24.10.1996

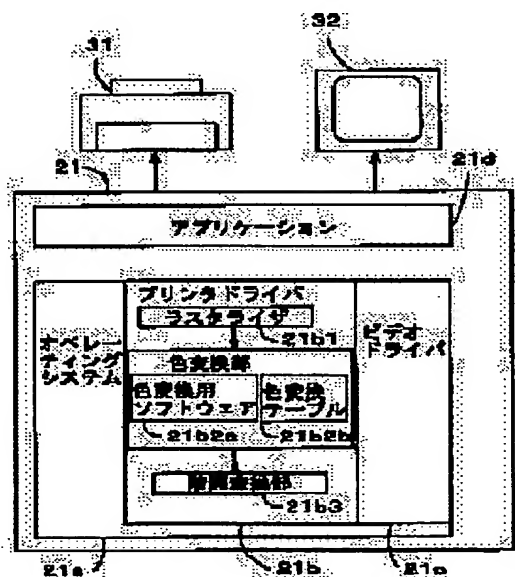
Priority country : JP

(54) APPARATUS AND METHOD FOR MANUFACTURING COLOR CONVERSION TABLE, AND RECORDING MEDIUM

(57)Abstract:

PROBLEM TO BE SOLVED: To generate a color conversion table optimum to the environment of the user or the like.

SOLUTION: When the installer is executed on a computer 2 being a component of an image processing unit, a color conversion table 21b2b is generated from an original color conversion table, number of grating points is increased by a nonlinear interpolation arithmetic operation or a linear interpolation arithmetic operation by utilizing the Lagrange's interpolation formula. In this case, number of the grating points may be fixed or number of grating points is selected in response to the environment or an input image and the color conversion table 21b2b with a proper size is generated from the original color conversion table with smaller size.



LEGAL STATUS

[Date of request for examination]

12.07.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision
of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

【特許請求の範囲】

【請求項1】 異なる表色空間の間で階調表色データを変換するために変換元の表色空間に複数の格子点を設定し、この格子点に変換先の表色空間での階調表色データを対応させた色変換テーブルを生成する色変換テーブルの製造装置であって、

少数の格子点において変換の対応関係を記憶する元色変換テーブルと、

この元色変換テーブルの格子点を補間演算によって増加させて変換に利用する色変換テーブルを生成する補間手段とを具備することを特徴とする色変換テーブルの製造装置。

【請求項2】 上記請求項1に記載の色変換テーブルの製造装置において、上記補間手段は、複数の格子点の対応関係から非線形補間演算で補間する非線形補間演算手段を具備することを特徴とする色変換テーブルの製造装置。

【請求項3】 上記請求項2に記載の色変換テーブルの製造装置において、上記元色変換テーブルは、均等な間隔の格子点となっていることを特徴とする色変換テーブルの製造装置。

【請求項4】 上記請求項1に記載の色変換テーブルの製造装置において、上記補間手段は、複数の格子点の対応関係から線形補間演算で補間する線形補間演算手段を具備することを特徴とする色変換テーブルの製造装置。

【請求項5】 上記請求項1～請求項4に記載の色変換テーブルの製造装置において、上記補間手段は、補間で増加させる格子点の数を選択可能としていることを特徴とする色変換テーブルの製造装置。

【請求項6】 上記請求項5に記載の色変換テーブルの製造装置において、上記補間手段は、増加させる格子点の数を環境に応じて設定することを特徴とする色変換テーブルの製造装置。

【請求項7】 上記請求項5に記載の色変換テーブルの製造装置において、上記補間手段は、増加させる格子点の数を変換画像の種類に応じて設定することを特徴とする色変換テーブルの製造装置。

【請求項8】 上記請求項1に記載の色変換テーブルの製造装置において、上記色変換テーブルはコンピュータにて参照するとともに、通常時は同コンピュータの補助記憶装置に上記元色変換テーブルを記憶するとともに、同色変換テーブルの参照実行時に上記コンピュータの主記憶領域に展開することを特徴とする色変換テーブルの製造装置。

【請求項9】 異なる表色空間の間で階調表色データを変換するために変換元の表色空間に複数の格子点を設定し、この格子点に変換先の表色空間での階調表色データを対応させた色変換テーブルを生成する色変換テーブルの製造方法であって、

少数の格子点において変換の対応関係を記憶する元色変

換テーブルの前記格子点を補間演算によって増加させて変換に利用する色変換テーブルを生成することを特徴とする色変換テーブルの製造方法。

【請求項10】 異なる表色空間の間で階調表色データを変換するために変換元の表色空間に複数の格子点を設定し、この格子点に変換先の表色空間での階調表色データを対応させた色変換テーブルをコンピュータにて生成する色変換テーブル作成プログラムを記録した記録媒体であって、

少数の格子点において変換の対応関係を記憶する元色変換テーブルの前記格子点を補間演算によって増加させて変換に利用する色変換テーブルを生成することを特徴とする記録媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、異なる表色空間の間で階調表色データを変換するために変換元の表色空間での格子点に変換先の表色空間での階調表色データを対応させた色変換テーブルを生成する色変換テーブルの製造装置および製造方法並びに色変換装置に関する。

【0002】

【従来の技術】従来、この種の色変換テーブルとして、コンピュータ上のカラー画像をカラー印刷するカラー印刷システムが知られている。

【0003】コンピュータの内部では、カラー画像は縦横に並べられた各画素ごとについて赤緑青の三原色

(R, G, B)で階調表示されているが、一般のカラー印刷装置においてはシアン、マゼンダ、イエローの三色(C, M, Y)あるいはこれにブラックを加えた(C, M, Y, K)四色で階調表示のない状態で印刷される。従って、カラー印刷するためには赤緑青の三原色(R, G, B)の表示からシアン、マゼンダ、イエローの三色(C, M, Y)の表示への変換の作業と、階調表示から階調のない表示への階調変換の作業が必要となる。なお、色空間自体は一つの空間であるものの、座標の取り方によって表示が異ならざるをえないため、以下においては、便宜上、座標の取り方に応じた表色空間と呼ぶことにする。

【0004】この(R, G, B)表示から(C, M, Y)表示への変換は変換式によって一義的に定まるものではなく、それぞれの階調を座標とする色空間について相互に対応関係を求めておき、この対応関係から逐次変換するのが通常である。ここにおいて、少なくとも変換元の(R, G, B)表示が各色について256階調であったとすれば、約1670万個(256×256×256)の要素の色変換テーブルを持たなければならない。

【0005】効率的な記憶資源の利用を考えた結果、すべての座標値についての対応関係を用意しておくのではなく、適当なとびとびの格子点について対応関係を用意

しておき、補間演算を併用するようにしている。すなわち、(R, G, B) 表色空間の中でのある座標の色について (C, M, Y) 表色空間の対応関係を求めるときには同座標を取り囲む格子点の対応関係を利用し、線形補間などの演算を経て同座標の対応関係を求めている。

【0006】このような色変換テーブルは、一般にプリンタドライバが備えており、色変換テーブルを含めたプリンタドライバ自体は個々のカラー印刷装置に対応して一つだけが提供されている。従って、色変換テーブルについても、記憶資源との対比から適当に定められた格子点の数に特定されていた。

【0007】

【発明が解決しようとする課題】上述した従来の色変換テーブルにおいては、プリンタドライバを提供する側が一般的記憶資源との対比に基づいて色変換テーブルを作成しているため、必ずしもユーザの環境によっては最適なものとは限らないという課題があった。すなわち、ユーザの環境によってはまだまだ大きいという場合もあるし、より大きなサイズとした方が良いという場合もあった。

【0008】さらに、色変換テーブルの大きさによって印刷品質にも差が生じるため、一定の大きさの色変換テーブルでは十分ではないという課題もあった。

【0009】本発明は、上記課題にかんがみてなされたもので、ユーザの環境などに最適な色変換テーブルを生成することが可能な色変換テーブルの製造装置および製造方法の提供を目的とする。

【0010】

【課題を解決するための手段】上記目的を達成するため、請求項1にかかる発明は、異なる表色空間の間で階調表色データを変換するために変換元の表色空間での格子点に変換先の表色空間での階調表色データを対応させた色変換テーブルを生成する色変換テーブルの製造装置であって、少数の格子点において変換の対応関係を記憶する元色変換テーブルと、この元色変換テーブルの格子点を補間演算によって増加させて変換に利用する色変換テーブルを生成する補間手段とを具備する構成としてある。

【0011】上記のように構成した請求項1にかかる発明においては、もともと存在しているのが格子点を少数とした元色変換テーブルであり、記憶資源をわずかしかなければならない。そして、補間手段はこの元色変換テーブルの格子点を補間演算によって増加させて色変換テーブルを生成し、この生成された色変換テーブルを色変換に利用する。

【0012】元色変換テーブルにおける格子点の数は、生成される色変換テーブルと比較して相対的に少なければよく、必ずしも極めて少ない数値である必要はない。特に、三次元の格子点であれば、格子点の多少は3乗で変化してくる。すなわち、格子点を半分にするだけで色

変換テーブルのサイズは1/8の大きさとなり、少しの差でも記憶資源に対して大きな効果を与えることになる。

【0013】補間演算手段については各種の演算を適用することが可能であり、その一例として、請求項2にかかる発明は、請求項1に記載の色変換テーブルの製造装置において、上記補間手段は、複数の格子点の対応関係から非線形補間演算で補間する非線形補間演算手段を具備する構成としてある。

10 【0014】上記のように構成した請求項2にかかる発明においては、補間手段の非線形補間演算手段が複数の格子点の対応関係から非線形補間演算で格子点を補間する。

【0015】非線形補間演算を行うことにより、増加される格子点での対応関係が正確に再現されることになる。従って、元色変換テーブルの格子点を少なくとも非常に再現性の良い結果を得ることが可能となる。

20 【0016】一方、元色変換テーブルの格子点の間隔についてはこの補間演算にも関連があり、その一例として、請求項3にかかる発明は、請求項2に記載の色変換テーブルの製造装置において、上記元色変換テーブルは、均等な間隔の格子点となった構成としてある。

【0017】非線形補間演算においては必ずしも格子点間隔が均等である必要はないが、不均等な格子点間隔であるとする演算式の係数が複雑になってくる。そして、少なくとも三次元の立方体内で補間演算で格子点を増加させようとする場合には、各軸方向毎に複数の格子点から中間的な格子点を求めるような作業が必要になり、このような場合に演算式の係数が複雑になってくると補間演算が煩雑になり、作業が煩わしい。これに対して均等な格子点間隔である場合には一部の係数などが一定となってループ処理も適用しやすくなるなどの状況が生じる。

【0018】むしろ、非線形補間演算以外の他の補間演算によっては均等である方が好ましい場合もあるし、不均等である方が好ましい場合もあり、補間演算に応じて適宜変更可能である。

【0019】また、非線形補間演算以外の演算を利用する一例として、請求項4にかかる発明は、請求項1に記載の色変換テーブルの製造装置において、上記補間手段は、複数の格子点の対応関係から線形補間演算で補間する線形補間演算手段を具備する構成としてある。

【0020】線形補間の場合には演算式自体が複雑でないというメリットもあるし、演算に必要な格子点が軸方向に対して二点であるという性質がある。従って、対応関係が大きく変化する部分においては格子点を細かくすることにより、正確な対応関係を容易に得られるようになるし、逆に、対応関係がさほど変化しない部分においては格子点を粗くするといったことも容易になる。

50 【0021】補間演算手段が増加させる格子点の数は、

必ずしも一定である必要はなく、その一例として、請求項5にかかる発明は、請求項1～請求項4に記載の色変換テーブルの製造装置において、上記補間手段は、補間で増加させる格子点の数を選択可能とした構成としてある。

【0022】増加される格子点の数により色変換テーブルのサイズが変化するし、補間演算によっては格子点の色変換での変換精度に影響を与える場合もある。従って、補間で増加させる格子点の数が選択可能であることにより、ユーザの環境に対して最適な格子点の数とすることが可能となってくる。

【0023】この場合において、請求項6にかかる発明は、請求項5に記載の色変換テーブルの製造装置において、上記補間手段は、増加させる格子点の数を環境に応じて設定する構成としてある。

【0024】格子点の数によって色変換テーブルのサイズが変化するし、色変換時のヒット率も変わってくる。さらには、増加させる格子点の位置によっては補間演算を容易にさせたりすることにもなる。従って、環境に応じて総合的に格子点を増加させることにより、最適な色変換を提供できる。環境に応じて増加する格子点を選択する方針として、記憶資源が多ければ大きなサイズの色変換テーブルとすればよいし、格子点が多いほど色変換精度が高い場合において色変換精度を高くすることを望むのであれば格子点を多くするようにすればよいし、補間演算時に2の累乗での乗除算が容易であるようならば、格子点の間隔が2の累乗となるようにすればよい。

【0025】さらに、同様に増加する格子点を選択するにあたり、請求項7にかかる発明は、請求項5に記載の色変換テーブルの製造装置において、上記補間手段は、増加させる格子点の数を変換画像の種類に応じて設定する構成としてある。

【0026】上述したように、格子点が多いほど色変換精度が高い場合があり、そのような場合において画像によっては色変換精度を高く要求するものもあればさほど必要としないようなものもある。このため、変換画像の情報が色変換精度を高く要求するようなものであれば格子点をできるだけ多くし、色変換精度をさほど要求しないようなものであれば格子点をさほど多くしない。この場合の方針として、例えば、オペレーティングシステムなどから変換画像の種類を知ることができるような場合に、ファイルの拡張子がビットマップであれば写真などの色情報の重要度が高いものと判断して格子点をできる限り多くする一方、ファイルの拡張子がドローデータやビジネスグラフを指すような場合には色情報の重要度は低いものと判断して格子点をさほど増やさないといったことが有効である。

【0027】発明の思想の具現化例における他の一例として、請求項9にかかる発明は、異なる表色空間の間で階調表色データを変換するために変換元の表色空間に複

数の格子点を設定し、この格子点に変換先の表色空間での階調表色データを対応させた色変換テーブルを生成する色変換テーブルの製造方法であって、少数の格子点において変換の対応関係を記憶する元色変換テーブルの前記格子点を補間演算によって増加させて変換に利用する色変換テーブルを生成することを特徴としておる。

【0028】すなわち、必ずしも実体のある装置に限らず、その方法としても有効であることに相違はない。

【0029】ところで、このような色変換テーブルを備える色変換装置は単独で存在する場合もあるし、ある機器に組み込まれた状態で利用されることもあるなど、発明の思想としてはこれに限らず、各種の態様を含むものである。従って、ソフトウェアであったりハードウェアであったりするなど、適宜、変更可能である。

【0030】その一例として、印刷インクに対応した表色空間に対して異なる表色空間の階調表色データを変換するにあたり、少数の格子点において変換の対応関係を記憶する元色変換テーブルを使用し、格子点を補間演算によって増加させて色変換テーブルを生成し、この生成した色変換テーブルを利用して色変換する構成とするこ

ともできる。

【0031】すなわち、プリンタドライバは印刷インクに対応した表色空間に対して異なる表色空間の階調表色データを変換するために、色変換テーブルを参照することになるが、この際に少数の格子点の元色変換テーブルから補間して格子点を増し、格子点を増加した色変換テーブルを使用して色変換する。

【0032】発明の思想の具現化例として色変換装置のソフトウェアとなる場合には、請求項10のように、かかるソフトウェアを記録した記録媒体上においても当然に存在し、利用されるといわざるをえない。むしろ、その記録媒体は、磁気記録媒体であってもよいし光磁気記録媒体であってもよいし、今後開発されるいかなる記録媒体においても全く同様に考えることができる。また、一次複製品、二次複製品などの複製段階については全く問う余地無く同等である。その他、ソフトウェアである場合にインストール作業において上述したような格子点を増加させる処理を行うことも可能であるし、供給方法として通信回線を利用して行なう場合でも本発明が利用されていることにはかわりないし、提供する側はソフトウェア提供装置として機能するものであり、同様に本発明を利用していることに相違ない。

【0033】さらに、一部がソフトウェアであって、一部がハードウェアで実現されている場合においても発明の思想において全く異なるものではなく、一部を記録媒体上に記憶しておいて必要に応じて適宜読み込まれるような形態のものとしてあってもよい。さらには、かかる色変換テーブル使用することになるカラーファクシミリ機やカラーコピー機においても適用可能であることはいうまでもない。

【0034】

【発明の効果】以上説明したように本発明は、小さなサイズの元色変換テーブルから格子点を増した色変換テーブルを生成するものであるため、色変換をしない状態では最低限の記憶資源しか必要としない一方、色変換を行うときには必要な大きさとなった色変換テーブルとすることができ、より融通性の高い色変換テーブルの製造装置を提供することができる。むろん、必要なときにだけ展開して不要ときには展開しないようにしても良いし、記憶資源に余裕があるのであれば展開した状態のまま残しておくこともできる。

【0035】また、請求項2にかかる発明によれば、非線形補間演算で格子点を増加させるようにしているため、増加された格子点の精度が高くなり、良好な色変換結果を得ることができる。これは、逆にいえばより格子点を少なくしても良好な結果を得ることができるという効果にもなる。

【0036】さらに、請求項3にかかる発明によれば、元色変換テーブルの格子点を均等な間隔とすることにより非線形補間演算が複雑化しないようにでき、演算時間などの減少を図ることができる。

【0037】さらに、請求項4にかかる発明によれば、演算量の少ない線形補間を利用して簡易に補間演算を行うことができる。また、線形補間の簡易さを利用すれば大きな変化部分で格子点を密にすることにより、簡易でありながら精度の向上を図ることができるなどのメリットもある。

【0038】さらに、請求項5にかかる発明によれば、増加させる格子点の数を選択できるので、ユーザの記憶資源に合わせて色変換テーブルのサイズを決めるなど、より柔軟にユーザの環境に適した色変換テーブルを生成することができるようになる。これはまた、複数のサイズの色変換テーブルを生成することもできるので、必要に応じて適宜選択して利用するといったことも可能となる。

【0039】さらに、請求項6にかかる発明によれば、環境に応じた数の格子点を選択するので、ユーザが煩わしい設定作業を行わなくても良くなる。

【0040】さらに、請求項7にかかる発明によれば、格子点の増加を変換画像に応じて選択するようにしているため、不要に色変換テーブルを大きくしすぎたり、不十分にしか色変換テーブルを大きくしていないなどといった不具合を無くすることができる。

【0041】さらに、請求項9にかかる発明によれば、上述したのと同様に、より融通性の高い色変換テーブル生成することが可能な色変換テーブルの製造方法を提供することができる。

【0042】さらに、請求項19にかかる発明によれば、上述したのと同様に、より融通性の高い色変換テーブル生成することが可能な色変換テーブルの製造プログ

ラムを、コンピュータ等で読み取り可能に記録した記録媒体として提供することができる。

【0043】

【発明の実施の形態】以下、図面にもとづいて本発明の実施形態を説明する。

【0044】図1は、本発明の一実施形態にかかる画像処理システムをブロック図により示しており、図2は具体的ハードウェア構成例をブロック図により示している。

10 【0045】同図において、画像入力装置10はカラー画像を撮像するなどして階調表色データを画像処理装置20へ出力し、同画像処理装置20は所定の画像処理を行なって画像出力装置30に出力し、同画像出力装置30は元のカラー画像を表示する。

【0046】ここにおいて、画像入力装置10の具体例はスキャナ11やデジタルスチルカメラ12などが該当し、画像処理装置20の具体例はコンピュータ21とハードディスク22などからなるコンピュータシステムが該当し、画像出力装置30の具体例はプリンタ31やディスプレイ32等が該当する。また、本発明をコンピュータ等に実施させるプログラムを記録可能な記録媒体は、ドライブ装置23によってコンピュータに読み込まれるCD-ROM24等の記録媒体が相当する。

【0047】画像入力装置10としてのスキャナ11が階調表色データとして例えばRGB（緑、青、赤）の階調データを出力するものとするとともに、画像出力装置30としてのプリンタ31は階調表色データとしてCMY（シアン、マゼンダ、イエロー）の二値データを入力として必要とするものとする、画像処理装置20としてのこのコンピュータ21の具体的役割は、RGBの階調データをCMYの二値データに変換することである。また、ディスプレイ32がRGBの階調データを入力するものとしても、スキャナ11とディスプレイ32では色特性が異なるのが通常であり、コンピュータ21はRGBの階調データをRGBの階調データに変換する処理を行なうことになる。デジタルスチルカメラ12についてもほぼ同様のことがいえる。

【0048】このコンピュータ21の内部で行なわれる処理を図3に示している。図に示すように、コンピュータ21内ではオペレーティングシステム21aが稼働しており、プリンタ31やディスプレイ32に対応したプリンタドライバ21bやビデオドライバ21cが組み込まれている。一方、アプリケーション21dはオペレーティングシステム21aにて処理の実行を制御され、必要に応じてプリンタドライバ21bやビデオドライバ21cと連携して所定の処理を実行する。

【0049】アプリケーション21dで生成される印刷用データはオペレーティングシステム21aを介してプリンタドライバ21bに入力され、当該プリンタドライバ21bはプリンタ31が要求するフォーマットの画像

データに変換する。この変換が上述したRGBの階調データをCMYの二値データに変換する処理に該当する。ここにおいて、同プリンタドライバ21bは、アプリケーション21dが所定の画面単位で生成する画像データからプリンタ31における印刷ヘッドの走査範囲を切り出すラスライザ21b1と、この走査範囲の各画素について色変換テーブルを参照してRGBの階調データをCMYの階調データに変換する色変換部21b2と、CMYの階調データを二値データに階調変換する階調変換部21b3とから構成されている。なお、アプリケーション21dが生成する表示画像データについてはビデオドライバ21cが所定の画面用メモリに書き込み、ハードウェア回路を介してディスプレイ32にて表示させている。

【0050】色変換部21b2は色補正モジュールとも呼ばれ、色変換の演算処理を実行する色変換用ソフトウェア21b2aと色変換テーブル21b2bとから構成されている。色変換テーブル21b2bは異なる表色空間の間で階調表色データを変換するために変換元の表色空間での格子点に変換先の表色空間での階調表色データを対応させたものであり、より具体的には図4に示すような三次元のRGB階調データを座標値としてCMY階調データを読み出すための三次元ルックアップテーブルである。そして、色変換用ソフトウェア21b2aは各画素のRGB階調データを座標値としてCMY階調データを読み出す処理を行う。

【0051】この色変換部21b2を含めて同プリンタドライバ21bは、図5に示すインストールプログラムによってハードディスク22上に展開される。このインストーラは、機器チェックを行なうステップS110と、上記色変換用ソフトウェア21b2aを含むドライバ用ソフトウェアをハードディスク22上に展開するステップS120と、所定の補間演算によって小サイズの*

$$P(X) = \sum_{i=0}^{N-1} \left(Y_i \prod_{j \neq i} \left((X - X_j) / (X_i - X_j) \right) \right)$$

【0059】がラグランジュ (Lagrange) の補間公式である。なお、右辺の \prod 以下については $((X - X_j) / (X_i - X_j))$ を $j = i$ 以外の全ての j について掛け合わせたものを意味する。この補間演算の具体的な実行方法をC言語で示したコーディングリストを図6に示している。

【0060】さて、このような補間演算を用いれば、図7に示すような実際には各軸方向に五つの格子座標しか持たない小サイズの元色変換テーブル21b2cを使用して各格子座標間に新たな三つの格子座標を増加して格子座標間を四分することは可能である。この場合の格子座標間隔は「64」階調に相当し、元色変換テーブル21b2cの各格子座標に格子番号「0」～「4」を付すものとする。また、新たに格子座標間を四分することに

*元色変換テーブル21b2cから所定のサイズの色変換テーブル21b2bを生成するステップS130とから構成されている。

【0052】すなわち、補間演算によって小サイズの元色変換テーブル21b2cから所定のサイズの色変換テーブル21b2bを生成するステップS130こそが本発明における色変換テーブルの製造装置を構成し、その手順が色変換テーブルの製造方法を構成している。この具体的手法については後述するとして、本実施形態においては、プリンタドライバ21bのインストーラとして具現化されているものの、その機能として小サイズの元色変換テーブル21b2cから所定のサイズの色変換テーブル21b2bを生成するものであればよい。従って、色変換テーブルを単独で生成するソフトウェアであってもよいし、あるいはワイヤーロジックからなるハードウェアなどで構成することも可能である。また、後述するように色変換用ソフトウェア21b2aが必要に応じて色変換テーブル21b2bを生成するよう構成することも可能である。

【0053】次に、この補間演算処理について詳述する。

【0054】まず、補間演算処理の一例として非線形補間演算を採用する場合について説明する。

【0055】 n 個の点 (X_i, Y_i) ($i = 0, 1, \dots, n-1$) が与えられれば $Y_i = P(X_i)$ ($i = 0, 1, \dots, n-1$) を満たす $n-1$ 次の多項式

【0056】

【数1】

$$P(X) = c_{n-1} \cdot X^{n-1} + c_{n-2} \cdot X^{n-2} + \dots$$

【0057】が一意的に定まる。但し、どの二つの X_i も等しくないとする。この多項式を表す閉じた式

【0058】

【数2】

より色変換テーブル21b2bでは格子番号「0」～「16」となり、格子座標間隔は「16」階調になる。なお、本来「256」階調としてこのように均等に分配する格子点を設けるのは不可能であり、計算上は「0」～「256」というような(実際の階調数+1)の格子番号を想定しておき、計算の最後で最後の格子番号(例えば256)を現実の階調の範囲の最後の格子番号(255に相当)に移行させることによって計算を簡易にする。

【0061】元色変換テーブル21b2cのインストーラプログラム内での記憶フォーマットについて概略を触れておくと、図8の上段に示すように、RGBの各成分を座標値としてそれぞれCMYの三色のデータを対応すべく、要素数が(5, 5, 5, 3)の配列となってお

り、ファイルの先頭からベタで書き込まれている。従って、元色変換テーブル21b2cの対応データを参照するためにはR軸とG軸とB軸とのそれぞれに対応した格子番号ポインタPr, Pg, Pbを設定し、ファイルの先頭から $(Pr \times 5 \times 5 \times 3 + Pg \times 5 \times 3 + Pb \times 3)$ をオフセットアドレスとしてシアン(C)は「1」バイト目、マゼンダ(M)は「2」バイト目、イエロー(Y)は「3」バイト目を読み出すことになる。

【0062】むしろ、かかる記憶フォーマットは一例に過ぎず、例えば、シアンについて全座標分だけ並べ、マゼンダ、イエローについて、順次、全座標分だけ並べていくなどの配置でも構わない。あるいは、ファイル圧縮した状態で保存してあっても良い。ただし、配列としてベタで書き込まれている場合には、後述するようにポインタ値で読み出しアドレスを演算でき、読み出す際の規則性を自由に設定できる。

【0063】一方、色変換テーブル21b2bの記憶フォーマットを図8の下段に示している、上述したように元色変換テーブル21b2cの各格子点間を四分して新たに三つの格子座標を形成するものであるため、要素数が(17, 17, 17, 3)の配列となっており、ファイルの先頭からベタで書き込むようにしている。従って、この色変換テーブル21b2bの対応データを参照するためには先程と同様の格子番号ポインタPr, Pg, Pbを設定すると、ファイルの先頭から $(Pr \times 17 \times 17 \times 3 + Pg \times 17 \times 3 + Pb \times 3)$ をオフセットアドレスとしてシアン(C)は「1」バイト目、マゼンダ(M)は「2」バイト目、イエロー(Y)は「3」バイト目を読み出すことになる。

【0064】色変換テーブル21b2bの格子点を増加する処理を図9のフローチャートに示している。この処理では、図8に示すように元色変換テーブル21b2cと色変換テーブル21b2bとで一致する格子点が存在するので、ステップS210にて格子点データを移転させる作業を行いつつ、格子点の間に空白のデータを挿入してハードディスク22上にファイルの形として展開する。この後、ステップS220にて色変換テーブル21b2bの全格子点について対応データを埋めるべく、各軸の格子番号についてネストしたループ処理を実行する。格子番号は「0」～「16」であるので、R軸、G軸、B軸についてポインターに「0」～「16」を設定して処理を繰り返す。最も内側のループ内ではR軸、G軸、B軸のポインターで示される格子点が元色変換テーブル21b2cから移行した格子点に一致するか否かを判断し、一致しなければ補間演算で格子点の対応データを算出する処理を実行する。しかし、一致するのであれば既に対応データがあるので補間演算の処理をスキップする。

【0065】補間演算の処理の一例を、図10～図12に示している。まず、フローを説明する前に図11にて

図示した非線形演算の概念を説明する。

【0066】数2に示したラグランジュの補間公式を四点の対応データに基づいて適用しようとした場合、図11に示すP点(Rp, Gp, Bp)の補間演算を行なうこととしても、必ずしも四つの格子点を通過するかどうかは不明である。従って、P点が位置する前後で各軸方向に四つの格子点からなる立方体を想定し、この立方体内で各軸方向ごとに順に補間演算を実行することにより、P点の演算に必要な四つの点の対応データを算出していくことにする。ここにおいて、各軸毎の格子座標を{R1, R2, R3, R4} {G1, G2, G3, G4} {B1, B2, B3, B4}と設定しておく。

【0067】まず、P点(図示△の点)を通過するG軸方向に平行な直線を想定すると、この直線は、G軸の格子座標を通過することになる四つのRB平面を貫通することになる。この各交点は同図にて○点で示しており、その座標は(Rp, G4, Bp)、(Rp, G3, Bp)、(Rp, G2, Bp)、(Rp, G1, Bp)である。この交点自体の対応データは不明であるため、それぞれの交点と交わるRB平面上でB軸に平行な直線を想定する。この直線はB軸の格子座標を通過することになる四つのRG平面を貫通する。四つの直線のうちG軸の座標が「G1」である点に注目し、各交点を同図にて●点で示している。その座標は(Rp, G1, B1)、(Rp, G1, B2)、(Rp, G1, B3)、(Rp, G1, B4)であり、まだ対応データは不明である。しかしながら、これらの交点を通過するR軸に平行な直線を想定すると、今度は全て格子点を通過する。すなわち、交点(Rp, G1, B1)を通過する直線は(R1, G1, B1)、(R2, G1, B1)、(R3, G1, B1)、(R4, G1, B1)を通過する。

【0068】これを逆に遡ることにすれば、四つの(R1, G1, B1)、(R2, G1, B1)、(R3, G1, B1)、(R4, G1, B1)から一つの●点の対応データを得ることができることになり、同様にして四つの●点の対応データを得たときには一つの○点の対応データを得ることができる。これを繰り返せば四つの○点の対応データを得ることができ、そうすれば△点の対応データを算出できるようになる。

【0069】かかる過程のより具体的な演算を図12に示しており、一番内側のネストでは $i = 1 \sim 4$ とした四つの格子点の対応データD(Ri, Gj, Bk)を利用し、R軸方向での成分値Rpでの対応データf(j)

(●点の対応データ)を算出している。j = 1 ~ 4として四つのf(j)が得られれば、一つ上のネスト内ではこれを利用してB軸方向での成分値Bpでの対応データg(k)(○点の対応データ)を算出する。そして、k = 1 ~ 4として四つのg(k)が得られれば、一番上のネスト内ではこれを利用してh(△点の対応データ)が算出できる。

【0070】図10に示すフローチャートに戻ると、ステップS310では所属格子グループの特定を実行する。図11及び図12に示すように各軸方向に四つの格子点を固定して演算を実行すると容易であるため、演算のルーチンをこの立方体の座標値を利用して実行できるサブルーチン化している。従って、格子点を補間する演算を実行する前に当該格子点を含むような各軸方向に四つの格子点からなる立方体を特定する。そして、ステップS320ではこの立方体の格子点における対応データを同ワークエリアへ移動させる。

【0071】ワークエリアでは図11に示す関係が特定されるため、続くステップS330では図12に示すネスト処理で非線形演算を実行する。なお、ワークエリアへ移動させる際には各軸方向へのオフセットが生じるため、移動させる際にオフセット量を保存し、増加する格子点についてもその座標値に同オフセット量を考慮した座標値(Rp, Gp, Bp)で計算する。なお、図12においては三次元での補間に対応して三段階のネストの処理となっているが、さらに高次元での補間に対応してネスト処理することも可能である。

【0072】このようにして元色変換テーブル21b2cの格子点以外の格子点で非線形演算を施していくことにより、各軸のループ処理を終了したときに完全な色変換テーブル21b2bを得ることができる。

【0073】上述した実施形態においては、非線形演算の具体的な処理としてラグランジュの補間公式を利用しているが、他の演算を利用することも可能であり、例えば、スプライン(spline)補間も可能である。スプライン補間は導関数まで連続性を有する利用ができ、この意味で導関数の連続性が問題となる場合に備えた硬めの補間である。ただし、計算は複雑とならざるを得ず、このスプライン補間演算の具体的な実行方法をC言語で示したコーディングリストを図13に示している。

【0074】また、他の非線形の補間演算として、ネビル(Neville)補間であったり、ニュートン(Newton)補間などを利用可能である。これらの場合は、数値的にも計算が楽になる。

【0075】一方、上述した格子点については格子間隔を一定としている。これにより、特定の立方体をワークエリアとして対応データを移動させてしまい、演算を分かりやすく実行することができるようになる。しかしながら、必ずしも格子間隔が一定でなければならないわけではなく、格子間隔などを加味した係数などを利用して実行することも可能である。

【0076】さらに、上述した実施形態においては、ステップS220の補間演算処理で図10～図12に示す非線形補間を利用しているが、図14～図19に示すように線形補間を利用することも可能である。

【0077】図14には格子点を増加する前の格子点位置を白丸で示すとともに、格子点を増加した後の格子点

位置を黒丸で示しており、演算の簡易のため、格子間隔を半分にする位置に新たな格子点を設けている。従って、図に示す当初の格子番号は括弧書きのような丁度二倍の格子番号となる。なお、当初の格子点の数を仮に「i」として説明する。また、図15は線形補間の処理を実行するCPUの手順をフローチャートにより示しており、図16は最初の対応データの移動の状況を示しており、図17は補間される格子点を示しており、図18は補間演算の状況を示している。

10 【0078】格子点間を半分とする格子点を各軸に形成するものとする、補間前の格子点の格子座標は図14の括弧書きに示すように自動的に(0, 2, 4, 6, 8...)となり、その間を補間することになる。図15に示すフローチャートに戻ると、まず、CPUはステップS410にて既にテーブル内にある格子点データを新たなテーブルの所定位置に移行する処理を行う。例えば、図16に示すように、格子座標(0, 0, 0)の対応データは新たなテーブルの格子座標(0, 0, 0)の対応データとして、格子座標(0, 0, 1)の対応データは新たなテーブルの格子座標(0, 0, 2)の対応データとして、格子座標(0, 0, 2)の対応データは新たなテーブルの格子座標(0, 0, 4)の対応データとしてと

20 いうようにして移行していく。

【0079】線形補間で格子点を補間する場合、周囲の八つの格子点からなる格子立方体内の位置によって演算が異なる。すなわち、辺上に存在する格子点の場合は両側の二点の格子点から補間されるし、面上に存在する格子点の場合は周辺の四つの格子点から補間されるし、中心に存在するもの場合は八つの格子点から補間される。

30 【0080】格子点を増加する順序として、まず、ステップS420では格子辺上で格子点を生成する処理を実行する。CPUの演算処理では各軸毎にパラメータを与えてネストしたループで処理を行うため、図中においてもブロックを入れ子状に表示している。

【0081】パラメータは各軸ともに「0」、「2」、「4」、「6」、「8」...と与え、R軸方向について例えば格子座標(1, 0, 0)の対応データを格子座標(0, 0, 0), (2, 0, 0)のデータから生成する。即ち、図18に示すように、格子座標(0, 0, 0)の対応データX1と格子座標(2, 0, 0)の対応データX2とを足し、その結果X3を「2」で割ったもののX4となる。ここにおいて「2」の除算は二進数データにおいて1ビットの右シフトに対応し、極めて容易に実行できる。むしろ、最初に1ビットの右シフトを実行しておいてから足しても良く、この場合は演算過程でのオーバーフローを防止できる。以下、このパラメータの全組合せから格子辺上の格子点を生成する。

【0082】ステップS430では格子面上で格子点を生成する処理を実行する。この場合もネストしたループ

で処理を行うため、各軸のパラメータとして「0」、「2」、「4」、「6」、「8」と与え、RG面と平行な面についていえば格子座標(1, 1, 0)の対応データを格子座標(0, 0, 0), (0, 2, 0), (2, 0, 0), (2, 2, 0)のデータから生成する。この場合は四つの格子点の平均値を取ることになり、四つのデータを足してから「4」で割ればよい。なお、「4」の除算は二進数データにおいて2ビットの右シフトに対応し、極めて容易に実行でき、以下、このパラメータの全組合せから格子面上の格子点は生成される。

【0083】最後に、ステップS440では中心点の格子点を生成する処理を実行する。この場合は、各軸のパラメータとして「1」、「3」、「5」、「7」…と与え、格子座標(1, 1, 1)の対応データは周縁の八つの格子座標(0, 0, 0), (0, 0, 2), (0, 2, 0), (0, 2, 2), (2, 0, 0), (2, 0, 2), (2, 2, 0), (2, 2, 2)の対応データから生成する。この場合は八つの格子点の平均値を取ることになり、オーバーフローしないように3ビットの右シフトを実行してから足し合わせればよい。以下、このパラメータの全組合せから全中心点の格子点が生成される。

【0084】以上の処理を行うことによって格子点の補間が終了する。本実施形態においては格子間隔を半分にするように格子点を増加させているが、この例に限らず、必要に応じて適宜増減可能であり、記憶資源の許容範囲内で格子点を増加させればよい。

【0085】また、上述した線形補間演算でも格子点の格子間隔が一定となっているが、線形補間においては格子間隔を適宜変えたとしても演算の基本は両側の二点の対応データにしか過ぎないため、格子間隔を変えたとしても演算は容易である。従って、対応データの変化カーブが大きい部分では格子間隔を細かくすることにより、演算容易のまま補間精度を向上させることができる。

【0086】一方、非線形補間においても線形補間においても、増加させる格子点を必ずしも一定とする必要はなく、必要に応じて変化させることも可能である。

【0087】格子点を増加させる処理の一例として、図19のフローチャートには、システムの機器構成に応じてテーブル生成するシステム対応格子点増加処理を示している。

【0088】この例では、ステップS510にて演算能力を表すCPUの種類を入力を行ない、ステップS520にて同様に演算速度を表すクロックの入力を行い、ステップS530にて演算能力や演算速度に影響するメモリ容量の入力を行い、ステップS540にて生成先のハードディスクの残り容量を入力する。

【0089】そして、これらの組合せに対応して予め設定されているシステム対応テーブルをステップS550にて参照し、最も適切な格子点の数を読み出す。格子点

の数を得られたらステップS560にて格子点増加処理を行う。このシステム対応テーブルに記憶されている格子点の数は、一般的な傾向として、演算能力や演算速度が速ければ格子の間隔が大きくなり、ハードディスクの残り容量が多ければ格子の間隔は小さくなるといったように設定しておけばよい。むろん、機器構成の入力要素はこれらに限るものではないし、その軽重も一定ではない。例えば、ハードディスクの残り容量が多い場合、キャッシュとの兼ね合いもあるもののフルサイズのテーブルを作成することも不可能ではない。

【0090】この場合、ユーザの使用環境として印刷する対象が写真などのビットマップ系のデータが多いかあるいはドローデータ系のデータが多いかをステップS540とステップS550との間で問い合わせるようにしても良い。そして、ビットマップ系のデータが多いならば、写真などの色再現性に重きを置かれている環境を想定して格子点の数を大きくすれば良いし、ドローデータ系のデータが多いならばビジネスグラフなどの色再現性があまり重要でない環境を想定して格子点の数を小さくすれば良い。

【0091】これまではインストール時に格子点の増加処理を行って色変換テーブル21b2bを生成するようにしているが、印刷時に必要なサイズの色変換テーブル21b2bをコンピュータ21のRAM等の高速アクセス可能な記憶部に生成するようにしてもよい。通常時は小さな元色変換テーブル21b2cをハードディスクに格納しておき、印刷実行時に必要なサイズの色変換テーブル21b2bを高速アクセス可能なRAM上に作るメリットは非常に大きい。むろんこの場合もハードディスクの場合と同様に、RAMでの利用可能な残り容量や、出力する画像データの量、望まれる出力品質等を参酌しながら格子の間隔を設定して展開するサイズを決めればよい。

【0092】図3に示すように、アプリケーション21dが印刷する場合にはオペレーティングシステム21aを介してプリンタドライバ21bが起動されるが、このときにファイルタイプがプリンタドライバ21bに渡される。プリンタドライバ21bではこの時のファイルタイプ(例えば、bmpなど)からビットマップ系であるのかドローデータ系であるのかを判断し、それに対応した格子点の数を設定して色変換テーブル21b2bを生成する。この格子点の数の大きさについては上述したインストーラによる場合と同様の傾向で設定すればよい。むろん、入力データの種別を判別する方法としてはこのようなファイルタイプだけに限らず、実際の入力データの色数が多いか少ないかなどによって判断しても良いし、オペレーティングシステムがオブジェクトの種別を判別してプリンタドライバに通知するようにしても良い。

【0093】このように、画像処理装置20を構成する

コンピュータ21にてインストーラが実行されると、ステップS130で元色変換テーブル21b2cから色変換テーブル21b2bを生成するが、このときの格子点増加処理では、ステップS430におけるラグランジュの補間公式を利用した非線形補間演算で格子点を増加させたり、線形補間で格子点を増加させるなどし、また、その際に固定した格子点の数であっても良いし環境や入力画像に応じた格子点の数としてもよく、小さなサイズの元色変換テーブル21b2cから適切なサイズの色変換テーブル21b2bを生成することができる。

【図面の簡単な説明】

【図1】本発明の一実施形態にかかる色変換テーブルの製造装置を適用した画像処理システムのブロック図である。

【図2】同画像処理システムの具体的ハードウェア構成例のブロック図である。

【図3】コンピュータのソフトウェア構成を示すブロック図である。

【図4】三次元ルックアップテーブルの概念を示す図である。

【図5】インストールプログラムのフローチャートである。

【図6】ラグランジュの補間演算をC言語でコーディングした図である。

【図7】元色変換テーブルの格子座標を示す図である。

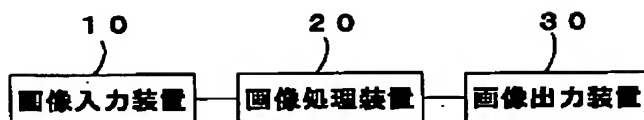
【図8】元色変換テーブルと色変換テーブルのファイル構成を示す図である。

【図9】格子点増加処理プログラムのフローチャートである。

【図10】非線形補間プログラムのフローチャートである。

【図11】ラグランジュの補間公式で非線形補間する場合の手順を示す概念図である。

【図1】



【図12】ラグランジュ補間演算に対応したフローチャートである。

【図13】スプライン補間演算をC言語でコーディングした図である。

【図14】格子点を増加する前後の格子座標を示す説明図である。

【図15】線形補間の格子点増加処理プログラムのフローチャートである。

【図16】元色変換テーブルと色変換テーブルのファイル構成を示す図である。

【図17】補間される格子点の位置を示す概略説明図である。

【図18】ビットシフトを併用した演算の状態を示す説明図である。

【図19】システム対応格子点増加処理プログラムのフローチャートである。

【符号の説明】

20…画像処理装置

21…コンピュータ

21a…オペレーティングシステム

21b…プリンタドライバ

21b1…ラスライザ

21b2…色変換部

21b2a…色変換用ソフトウェア

21b2b…色変換テーブル

21b2c…元色変換テーブル

21b3…階調変換部

21c…ビデオドライバ

21d…アプリケーション

22…ハードディスク

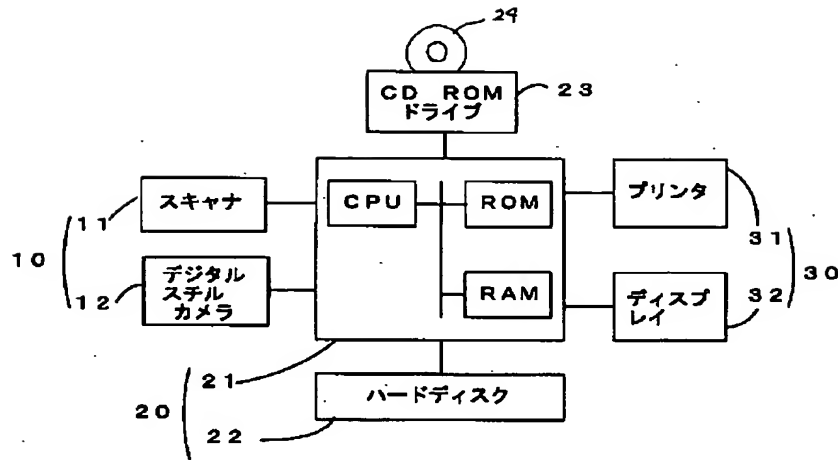
23…ドライブ装置

24…CD-ROM

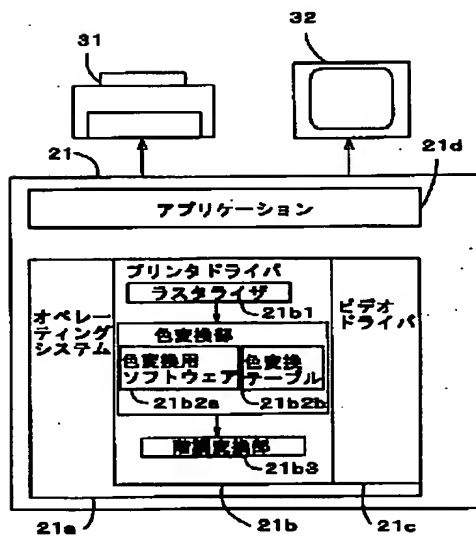
【図4】

Cx (Rx, Gx, Bx)
 Mx (Rx, Gx, Bx)
 Yx (Rx, Gx, Bx)

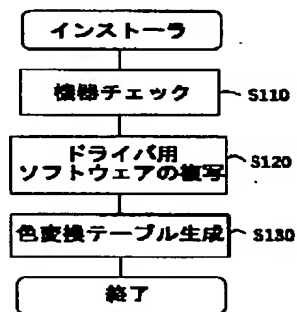
【図2】



【図3】



【図5】



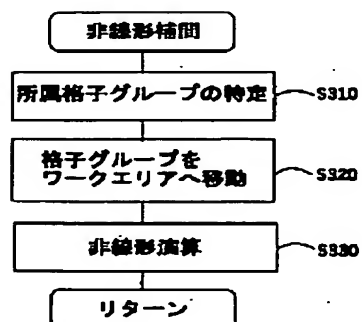
【図6】

```

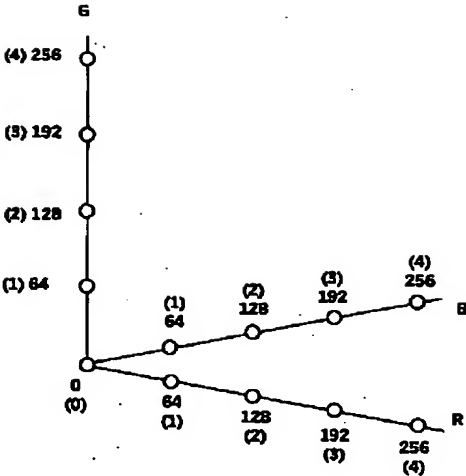
1 double lagrange(double t)
2 {
3     int i,j;
4     double sum,prod;
5
6     sum=0;
7     for (i=0;i<N;i++) {
8         prod=y[i];
9         for (j=0;j<N;j++)
10             if (j!=i) prod *= (t-x[j])/(x[i]-x[j]);
11         sum +=prod;
12     }
13     return sum;
14 }

```

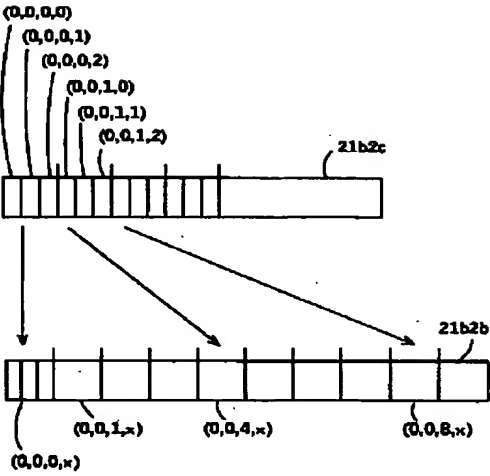
【図10】



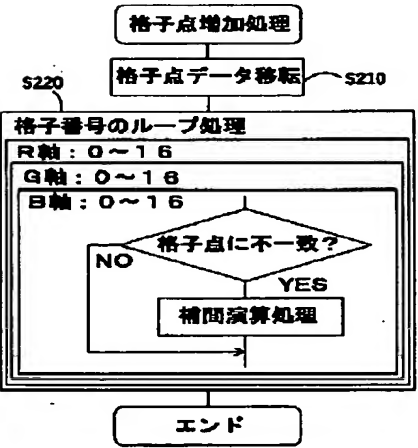
【図7】



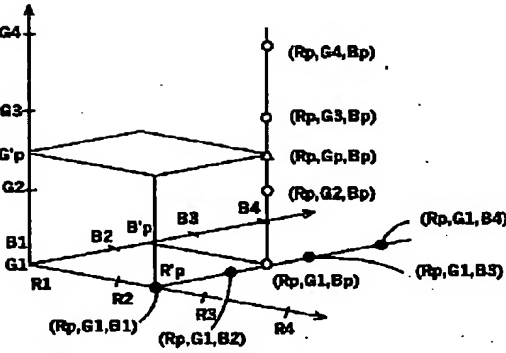
【図8】



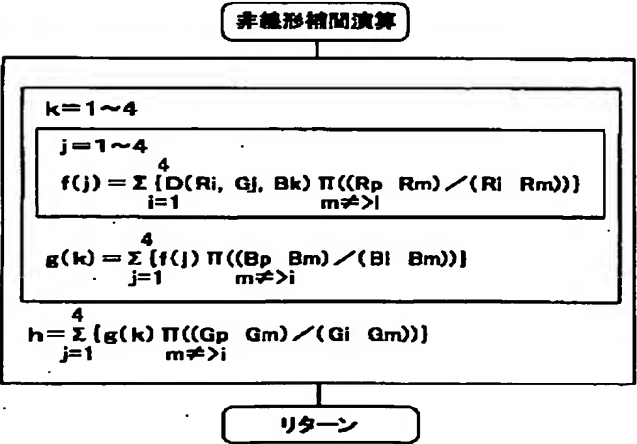
【図9】



【図11】



【図12】



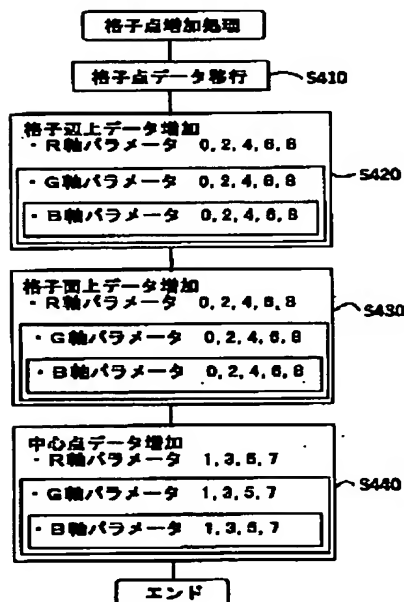
【図13】

```

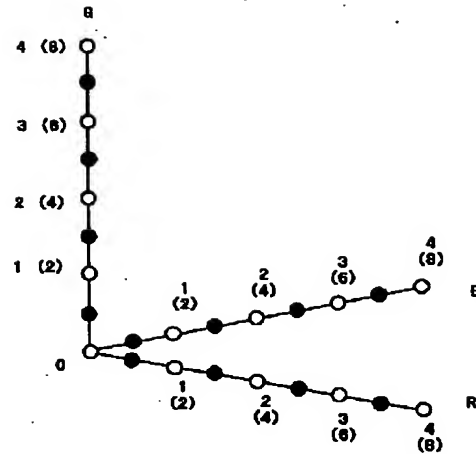
1 void maketable(double x[ ], double y[ ], double z[ ])
2 {
3     int i;
4     double t;
5     static double h[N], d[N];
6
7     x[0]=0; x[N-1]=0; /* 両端点でのy N(x)/8 */
8     for (i=0; i < N-1; i++) {
9         h[i] = x[i+1]-x[i];
10        d[i+1]=(y[i+1]-y[i])/h[i];
11    }
12    x[1]=d[2]-d[1]-h[0] * x[0];
13    d[1]=2 * (n[2]-x[0]);
14    for (i=1; i < N-2; i++) {
15        t=h[i]/d[i];
16        x[i+1]=d[i+2]-d[i+1]-x[i] * t;
17        d[i+1]=2 * (x[i+2]-x[i]-h[i] * t);
18    }
19    x[N-2]=h[N-2] * x[N-1];
20    for (i=N-2; i > 0; i--)
21        x[i]=(x[i]-h[i] * x[i+1])/d[i];
22 }
23
24 double spline(double t, double x[ ], double y[ ], double z[ ])
25 {
26     int i, j, k;
27     double d, h;
28
29     i=0; j=N-1;
30     while (i < j) {
31         k=(i+j)/2;
32         if (x[k] < t) i=k+1; else j=k;
33     }
34     h=(t > 0) ? t-x[i] : x[j]-t;
35     d=x[i+1]-x[i]; d+=x[j] * 3 * d;
36     return ((x[i+1]-x[i]) * d/h+x[i] * 3 * d
37            +(y[i+1]-y[i])/h
38            -(x[j] * 2 * x[i+1] * h)) * d+y[i];
39 }

```

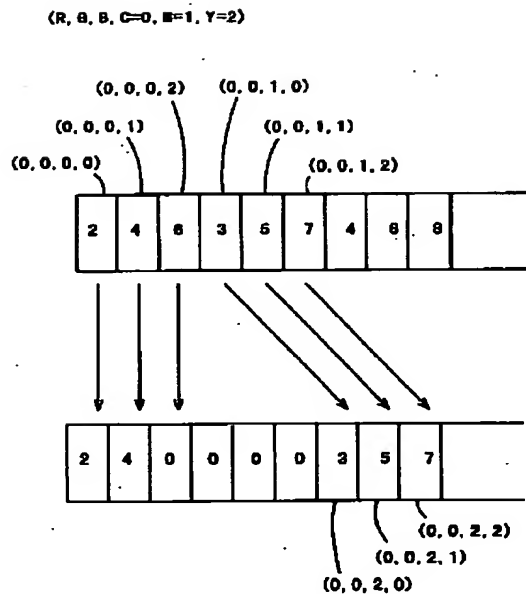
【図15】



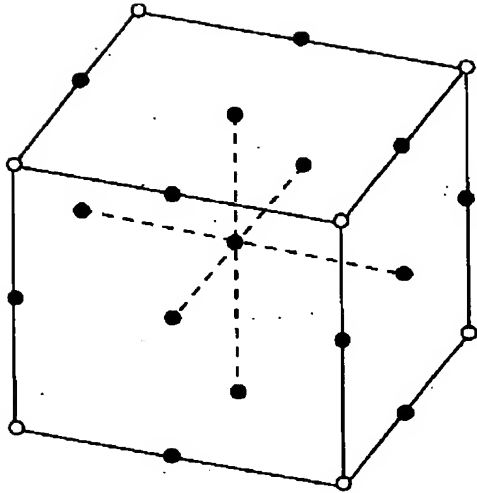
【図14】



【図16】



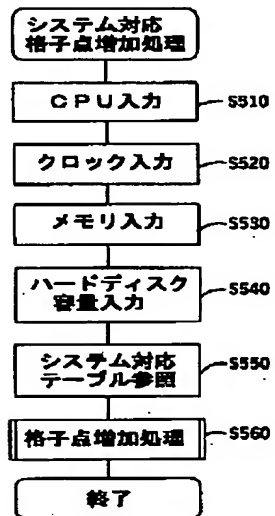
【図17】



【図18】

x1	0	0	1	0	1	0	0	0
+								
x2	0	0	1	0	0	0	1	0
//								
x3	0	1	0	0	1	0	1	0
$\div 2$ \rightarrow 1 bit								
x4	0	0	1	0	0	1	0	1

【図19】



フロントページの続き

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] In order to change gradation color specification data between different color specification space, two or more lattice points are set as the color specification space of a changing agency. The former color translation table which is the manufacturing installation of the color translation table which generates the color translation table which made the gradation color specification data in the color specification space of a conversion place correspond to this lattice point, and memorizes the response relation of conversion in a small number of lattice point, The manufacturing installation of the color translation table characterized by providing a interpolation means to generate the color translation table which is made to increase the lattice point of this former color translation table by the interpolation operation, and is used for conversion.

[Claim 2] It is the manufacturing installation of the color translation table characterized by providing a nonlinear interpolation operation means to interpolate the above-mentioned interpolation means by the nonlinear interpolation operation from the response relation of two or more lattice points in the manufacturing installation of a color translation table given in above-mentioned claim 1.

[Claim 3] It is the manufacturing installation of the color translation table characterized by the above-mentioned former color translation table serving as the lattice point of equal spacing in the manufacturing installation of a color translation table given in above-mentioned claim 2.

[Claim 4] It is the manufacturing installation of the color translation table characterized by providing a linear interpolation operation means to interpolate the above-mentioned interpolation means by the linear interpolation operation from the response relation of two or more lattice points in the manufacturing installation of a color translation table given in above-mentioned claim 1.

[Claim 5] It is the manufacturing installation of the color translation table characterized by making selectable the number of the lattice points which increase the above-mentioned interpolation means with interpolation in the manufacturing installation of a color translation table given in above-mentioned claim 1 - claim 4.

[Claim 6] It is the manufacturing installation of the color translation table characterized by setting up the number of the lattice points to which the above-mentioned interpolation means is increased in the manufacturing installation of the color translation table of a publication at above-mentioned claim 5 according to an environment.

[Claim 7] It is the manufacturing installation of the color translation table characterized by setting up the number of the lattice points to which the above-mentioned interpolation means is increased in the manufacturing installation of the color translation table of a publication at above-mentioned claim 5 according to the class of resolution picture.

[Claim 8] It is the manufacturing installation of the color translation table usually characterized by developing to the main storage area of the above-mentioned computer at the time of reference activation of a same color translation table while a computer refers the above-mentioned color translation table in the manufacturing installation of a color translation table given in above-mentioned claim 1 and memorizing the above-mentioned former color translation table to the auxiliary storage unit of this

computer at the time.

[Claim 9] In order to change gradation color specification data between different color specification space, two or more lattice points are set as the color specification space of a changing agency. It is the manufacture approach of the color translation table which generates the color translation table which made the gradation color specification data in the color specification space of a conversion place correspond to this lattice point. The manufacture approach of the color translation table characterized by generating the color translation table which is made to increase said lattice point of the former color translation table which memorizes the response relation of conversion in a small number of lattice point by the interpolation operation, and is used for conversion.

[Claim 10] In order to change gradation color specification data between different color specification space, two or more lattice points are set as the color specification space of a changing agency. It is the record medium which recorded the color translation table creation program which generates the color translation table which made the gradation color specification data in the color specification space of a conversion place correspond to this lattice point by computer. The record medium characterized by generating the color translation table which is made to increase said lattice point of the former color translation table which memorizes the response relation of conversion in a small number of lattice point by the interpolation operation, and is used for conversion.

[Translation done.]

*** NOTICES ***

Japan Patent Office is not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Field of the Invention] This invention relates to a color inverter at the manufacturing installation and the manufacture approach list of a color translation table which generate the color translation table which made the gradation color specification data in the color specification space of a conversion place correspond to the lattice point in the color specification space of a changing agency, in order to change gradation color specification data between different color specification space.

[0002]

[Description of the Prior Art] Conventionally, the color printing system which color-prints the color picture on a computer is known as this kind of a color translation table.

[0003] Although a gradation indication of the color picture is given with the three primary colors (R, G, B) of red-green blue about every pixel put in order in all directions inside the computer, it is printed in the condition that there is no gradation display by four (C, M, Y, K) colors which added black to three colors (C, M, Y) of cyanogen, MAZENDA, and yellow or this in the common color airline printer. Therefore, in order to color-print, the activity of the color conversion to the display of three colors (C, M, Y) of cyanogen, MAZENDA, and yellow and the activity of the gray scale conversion to the display without gradation from a gradation display are needed from the display of red-green blue in three primary colors (R, G, B). In addition, although the color space itself is one space, since a display cannot but change with how to take a coordinate, it will be called the color specification space according to how to take a coordinate for convenience below.

[0004] By the transformation, the color conversion to the display from this (R, G, B) display (C, M, Y) does not become settled uniquely, asks for response relation mutually about the color space which makes each gradation a coordinate, and, usually changes it from this response relation serially. In here, at least, about each color, if the display (R, G, B) of a changing agency was 256 gradation, it must have the color translation table of about 16,700,000 elements (256x256x256).

[0005] Response relation about all coordinate values is not prepared, but he prepares response relation about the suitable discontinuous lattice point, and is trying to use a interpolation operation together, as a result of considering utilization of an efficient storage resource. That is, when asking for the response relation of color specification space about the color (C, M, Y) of the existing coordinate in the inside of color specification (R, G, B) space, the response relation of the lattice point which encloses this coordinate is used, and it is asking for the response relation of this coordinate through operations, such as linear interpolation.

[0006] Generally the printer driver is equipped with such a color translation table, and only one is offered corresponding to the color airline printer of each [printer driver / including a color translation table / itself]. Therefore, it was specified as the number of the lattice points suitably set from comparison with a storage resource also about the color translation table.

[0007]

[Problem(s) to be Solved by the Invention] In the conventional color translation table mentioned above,

since the side which offers a printer driver was creating the color translation table based on comparison with a common storage resource, depending on a user's environment, the technical problem that it was not necessarily the optimal occurred. That is, also when saying that it is better to say that it is still large depending on a user's environment, and to consider as bigger size, it was.

[0008] Furthermore, since a difference arose also in printing quality with the magnitude of a color translation table, the technical problem that the color translation table of fixed magnitude was not enough also occurred.

[0009] This invention was made in view of the above-mentioned technical problem, and aims at offer of the manufacturing installation of the color translation table which can generate the optimal color translation table for a user's environment etc., and the manufacture approach.

[0010]

[Means for Solving the Problem] In order to attain the above-mentioned object, invention concerning claim 1 It is the manufacturing installation of the color translation table which generates the color translation table which made the gradation color specification data in the color specification space of a conversion place correspond to the lattice point in the color specification space of a changing agency in order to change gradation color specification data between different color specification space. It has considered as the configuration possessing a interpolation means to generate the former color translation table which memorizes the response relation of conversion in a small number of lattice point, and the color translation table which is made to increase the lattice point of this former color translation table by the interpolation operation, and is used for conversion.

[0011] In invention concerning claim 1 constituted as mentioned above, the former color translation table which made the lattice point the fraction exists from the first, and it has not deer needed the storage resource only. And a interpolation means makes the lattice point of this former color translation table increase by the interpolation operation, generates a color translation table, and uses this generated color translation table for color conversion.

[0012] The number of the lattice points in a former color translation table does not need to be not necessarily very few numeric values relatively as compared with the color translation table generated that what is necessary is just few. Especially, if it is the lattice point of three dimensions, some of lattice points will change by the cube. That is, the size of a color translation table serves as one eighth of magnitude only by making the lattice point into one half, and big effectiveness will be given to a storage resource also with some differences.

[0013] About a interpolation operation means, it is possible to apply various kinds of operations, and invention concerning claim 2 is considered as the configuration possessing a nonlinear interpolation operation means to interpolate the above-mentioned interpolation means by the nonlinear interpolation operation from the response relation of two or more lattice points, in the manufacturing installation of a color translation table according to claim 1 as the example.

[0014] In invention concerning claim 2 constituted as mentioned above, the nonlinear interpolation operation means of a interpolation means interpolates the lattice point by the nonlinear interpolation operation from the response relation of two or more lattice points.

[0015] By performing a nonlinear interpolation operation, the response relation in the lattice point increased will be reproduced by accuracy. Therefore, even if it lessens the lattice point of a former color translation table, it becomes possible to obtain a result with dramatically sufficient repeatability.

[0016] On the other hand, there is relation also in this interpolation operation about spacing of the lattice point of a former color translation table, and invention concerning claim 3 is considered as the configuration used as the lattice point of spacing with the above-mentioned equal former color translation table in the manufacturing installation of a color translation table according to claim 2 as that example.

[0017] Although lattice point spacing does not necessarily need to be equal in a nonlinear interpolation operation, supposing it is an unequal lattice spacing, the multiplier of operation expression will become complicated. And when you are going to make it increase the lattice point by the interpolation operation within the cube of three dimensions at least, and an activity which asks for the in-between lattice point

from two or more lattice points for every shaft orientations is needed and the multiplier of operation expression becomes complicated in such a case, a interpolation operation becomes complicated and an activity is troublesome. On the other hand, in being equal lattice point spacing, some multipliers etc. become fixed and situations, such as becoming easy to apply loop-formation processing, arise.

[0018] Of course, depending on other interpolation operations other than a nonlinear interpolation operation, the more equal one may be desirable, the more nearly unequal one may be desirable, and it can change suitably according to a interpolation operation.

[0019] Moreover, invention concerning claim 4 is considered as the configuration possessing a linear interpolation operation means to interpolate the above-mentioned interpolation means by the linear interpolation operation from the response relation of two or more lattice points, in the manufacturing installation of a color translation table according to claim 1 as an example using operations other than a nonlinear interpolation operation.

[0020] In the case of linear interpolation, there is also a merit that the operation expression itself is not complicated, and there is a property in which the number of the lattice points required for an operation is two to shaft orientations. Therefore, when response relation makes the lattice point fine in the part which changes a lot, exact response relation can be easily obtained now and it also becomes easy for response relation to make the lattice point coarse in the part which does not change so much at reverse.

[0021] The number of the lattice points to which a interpolation operation means makes it increase does not necessarily need to be fixed, and invention concerning claim 5 is considered as the configuration in which the above-mentioned interpolation means presupposed that it is selectable the number of the lattice points to which it is made to increase with interpolation as the example in the manufacturing installation of a color translation table according to claim 1 to 4.

[0022] The size of a color translation table may change with the number of the lattice points increased, and the lattice point may affect the conversion precision in color conversion depending on a interpolation operation. Therefore, that the number of the lattice points to which it is made to increase with interpolation is selectable enables it to consider as the number of the optimal lattice points to a user's environment.

[0023] In this case, invention concerning claim 6 is considered as the configuration which sets up the number of the lattice points to which the above-mentioned interpolation means is increased according to an environment in the manufacturing installation of a color translation table according to claim 5.

[0024] The size of a color translation table changes with the number of the lattice points, and the hit ratio at the time of color conversion also changes. Furthermore, it also becomes making a interpolation operation easy depending on the location of the lattice point to which it is made to increase. Therefore, the optimal color conversion can be offered by making the lattice point increase synthetically according to an environment. What is necessary is for what is necessary just to be made to make [many] the lattice point, if there are many storage resources and it desires the color translation table of big size, then to make color conversion precision high as a policy which chooses the lattice point which increases according to an environment, when color conversion precision is high so that it is good and there is much lattice point, and just to make it spacing of the lattice point become with a power of 2, if it becomes as [easily / at the time of a interpolation operation / the multiplication and division in a power of 2].

[0025] Furthermore, in choosing the lattice point which increases similarly, invention concerning claim 7 is considered as the configuration which sets up the number of the lattice points to which the above-mentioned interpolation means is increased according to the class of resolution picture in the manufacturing installation of a color translation table according to claim 5.

[0026] Color conversion precision may be high, and there are some some which require color conversion precision highly depending on an image in such a case, and there are some some which are not needed so much, so that there is much lattice point, as mentioned above. For this reason, if it seems that the information on an resolution picture requires color conversion precision highly, the lattice point will be made [many / as possible], and the lattice point will not be made [many / so] if it seems that color conversion precision is not required so much. While the significance of color information, such as

a photograph, will judge it as a high thing and will make [many / as possible] the lattice point as a policy in this case if the extension of a file is a bit map when the class of resolution picture can be known from an operating system etc., when the extension of a file points out draw data and business graph, as for the significance of color information, it is effective to judge it as a low thing and not to increase the lattice point so much.

[0027] As other examples in the example of embodiment of the thought of invention, invention concerning claim 9 In order to change gradation color specification data between different color specification space, two or more lattice points are set as the color specification space of a changing agency. It is the manufacture approach of the color translation table which generates the color translation table which made the gradation color specification data in the color specification space of a conversion place correspond to this lattice point. It is characterized by generating the color translation table which is made to increase said lattice point of the former color translation table which memorizes the response relation of conversion in a small number of lattice point by the interpolation operation, and is used for conversion.

[0028] That is, there is no difference not only in the equipment which not necessarily has a stereo but in being effective as the approach.

[0029] By the way, a color inverter equipped with such a color translation table contains not only this but various kinds of modes as thought of that it may be used in the condition of existing independently and having been included in a certain device, and invention. Therefore, it can change suitably that it is software or hardware etc.

[0030] In changing the gradation color specification data of color specification space which is different to the color specification space corresponding to printing ink as that example, the former color translation table which memorizes the response relation of conversion in a small number of lattice point is used, the lattice point can be made to be able to increase by the interpolation operation, a color translation table can be generated, and it can also consider as the configuration which carries out color conversion using this generated color translation table.

[0031] That is, although a color translation table will be referred to in order that a printer driver may change the gradation color specification data of different color specification space to the color specification space corresponding to printing ink, color conversion is carried out using the color translation table which interpolated from the former color translation table of a small number of lattice point, and increased the increase of the lattice point, and the lattice point on this occasion.

[0032] When becoming the software of a color inverter as an example of embodiment of the thought of invention, naturally it exists like claim 10 on the record medium which recorded this software, and it must be said that it is used. of course, the record medium may be a magnetic-recording medium, may be a magneto-optic-recording medium, and can completely be considered the same way in any record media developed from now on. Moreover, about duplicate phases, such as a primary replica and a secondary replica, it is equivalent without room to completely ask. In addition, when it is software, it is also possible to perform processing to which the lattice point which was mentioned above in the install activity is made to increase, and it is indifferent from the side which changes to this invention being used, thru/or is offered functioning as software offer equipment, even when carrying out as the supply approach using a communication line; and using this invention similarly.

[0033] Furthermore, a part is software, when the part is realized by hardware, there is nothing that is completely different in the thought of invention, and it may be made into the thing of a gestalt which memorizes the part on the record medium and is read suitably if needed. Furthermore, it cannot be overemphasized that it can apply also in this color facsimile machine and color copier that will carry out a color translation table activity.

[0034]

[Effect of the Invention] It can consider as the color translation table which became magnitude required when performing color conversion while only the minimum storage resource was needed in the condition of not carrying out color conversion since this invention was what generates the color translation table which increased the lattice point from the former color translation table of small size as

explained above, and the manufacturing installation of a color translation table with more high versatility can be offered. Of course, only when required, it develops, and when unnecessary, you may make it not develop and it can also leave with the condition of having developed when allowances were in the storage resource.

[0035] Moreover, according to invention concerning claim 2, since he is trying to make the lattice point increase by the nonlinear interpolation operation, the precision of the increased lattice point becomes high and a good color conversion result can be obtained. If it says reverse, even if this will lessen the lattice point more, it also becomes the effectiveness that a good result can be obtained.

[0036] Furthermore, according to invention concerning claim 3, by making the lattice point of a former color translation table into equal spacing, a nonlinear interpolation operation can be prevented from complicating and reduction in the operation time etc. can be aimed at.

[0037] Furthermore, according to invention concerning claim 4, a interpolation operation can be simply performed using linear interpolation with few amounts of operations. Moreover, if it uses simple [of linear interpolation], though it is simple by making the lattice point dense in a big change part, there is also a merit of being able to aim at improvement in precision.

[0038] Furthermore, since the number of the lattice points to which it is made to increase can be chosen according to invention concerning claim 5, the color translation table which fitted a user's environment more flexibly can be generated, such as deciding the size of a color translation table according to a user's storage resource. Since this can also generate the color translation table of two or more sizes, it also becomes possible of it to choose suitably and to use if needed, again.

[0039] Furthermore, according to invention concerning claim 6, since the lattice point of the number according to an environment is chosen, a user becomes unnecessary to carry out troublesome setting out.

[0040] Furthermore, according to invention concerning claim 7, since he is trying to choose the increment in the lattice point according to an resolution picture, the nonconformity of not enlarging a color translation table too much unnecessarily, or having enlarged the color translation table only at imperfection etc. can be abolished.

[0041] Furthermore, according to invention concerning claim 9, the manufacture approach of the color translation table in which the thing with more high versatility to do for color translation table generation is possible can be offered the same with having mentioned above.

[0042] Furthermore, according to invention concerning claim 19, the manufacture program of a color translation table in which the thing with more high versatility to do for color translation table generation is possible can be offered the same with having mentioned above as a record medium recorded possible [reading] by computer etc.

[0043]

[Embodiment of the Invention] Hereafter, the operation gestalt of this invention is explained based on a drawing.

[0044] Drawing 1 shows the image processing system concerning 1 operation gestalt of this invention with the block diagram, and drawing 2 shows the example of a concrete hardware configuration with the block diagram.

[0045] In this drawing, a picture input device 10 picturizes a color picture, and outputs gradation color specification data to an image processing system 20, this image processing system 20 performs a predetermined image processing, and outputs it to the image output unit 30, and this image output unit 30 displays the original color picture.

[0046] In here, as for the example of a picture input device 10, a scanner 11, a digital still camera 12, etc. correspond, the computer system which the example of an image processing system 20 becomes from a computer 21, a hard disk 22, etc. corresponds, and, as for the example of the image output unit 30, a printer 31 and display 32 grade correspond. Moreover, the record medium of the CD-ROM24 grade from which the record medium which can record the program which makes a computer etc. carry out this invention is read into a computer by drive equipment 23 corresponds.

[0047] While the scanner 11 as a picture input device 10 shall output the gradation data of RGB (green,

blue, red) as gradation color specification data. If the printer 31 as an image output unit 30 shall need the binary data of CMY (cyanogen, MAZENDA, yellow) as an input as gradation color specification data. The concrete role of this computer 21 as an image processing system 20 is changing the gradation data of RGB into the binary data of CMY. Moreover, usually a scanner 11 will differ in a color property from a display 32 also as that into which a display 32 inputs the gradation data of RGB, and a computer 21 will perform processing which changes the gradation data of RGB into the gradation data of RGB. It can say that the same is almost said of a digital still camera 12.

[0048] The processing performed inside this computer 21 is shown in drawing 3. As shown in drawing, within the computer 21, operating system 21a is working and printer driver 21b and video driver 21c corresponding to a printer 31 or a display 32 are incorporated. On the other hand, application 21d, activation of processing is controlled by operating system 21a, it cooperates with printer driver 21b or video driver 21c if needed, and predetermined processing is performed.

[0049] The data for printing generated by application 21d are inputted into printer driver 21b through operating system 21a, and the printer driver 21b concerned is changed into the image data of the format which a printer 31 requires. It corresponds to the processing which changes into the binary data of CMY the gradation data of RGB which this conversion mentioned above. In here, this printer driver 21b consists of a rasterizer 21b1 which starts the scanning zone of the print head in a printer 31 from the image data which application 21d generates per predetermined screen, color converter 21b2 which change the gradation data of RGB into the gradation data of CMY with reference to a color translation table about each pixel of this scanning zone, and a gradation converter 21b3 which carries out gray scale conversion of the gradation data of CMY to binary data. In addition, about the display-image data which application 21d generates, video driver 21c writes in the predetermined memory for screens, and it is made to display on a display 32 through hardware circuitry.

[0050] The color converter 21b2 is also called a color correction module, and consists of software 21b2a for color conversion which performs data processing of color conversion, and color translation table 21b2b. Color translation table 21b2b is a three-dimensions look-up table for making the gradation color specification data in the color specification space of a conversion place correspond to the lattice point in the color specification space of a changing agency, in order to change gradation color specification data between different color specification space, and reading CMY gradation data by making the RGB gradation data of three dimensions as more specifically shown in drawing 4 into a coordinate value. And software 21b2a for color conversion performs processing which reads CMY gradation data by making the RGB gradation data of each pixel into a coordinate value.

[0051] These printer driver 21b including this color converter 21b2 is developed on a hard disk 22 by the install program shown in drawing 5. This installer is constituted from a step S130 which generates color translation table 21b2b of predetermined size from former color translation table 21b2c of small size by step S110 which performs a device check, step S120 which develops the software for drivers containing the above-mentioned software 21b2a for color conversion on a hard disk 22, and the predetermined interpolation operation.

[0052] That is, by the interpolation operation, just step S130 which generates color translation table 21b2b of predetermined size constitutes the manufacturing installation of the color translation table in this invention from former color translation table 21b2c of small size, and the procedure constitutes the manufacture approach of a color translation table. What is necessary is just to generate color translation table 21b2b of predetermined size from former color translation table 21b2c of small size as that function in this operation gestalt, although embodied as an installer of printer driver 21b noting that it mentions later about this concrete technique. Therefore, you may be the software which generates a color translation table independently, or it is also possible to constitute from hardware which consists of wire logic. Moreover, it is also possible to constitute so that it may mention later and software 21b2a for color conversion may generate color translation table 21b2b if needed.

[0053] Next, this interpolation data processing is explained in full detail.

[0054] First, the case where a nonlinear interpolation operation is adopted as an example of interpolation data processing is explained.

[0055] if n points (X_i, Y_i) ($i = 0, 1, \dots, n-1$) are given -- $Y_i = P(X_i)$ -- ($i = 0, 1, \dots$, the n -primary polynomial [0056] that fills $n-1$)

[Equation 1]

$$P(X) = c_{n-1} \cdot X^{n-1} + c_{n-2} \cdot X^{n-2} + \dots$$

[0057] ** -- it becomes settled uniquely. However, suppose that no two $X_i(s)$ are equal. The closed formula showing this polynomial [0058]

[Equation 2]

$$P(X) = \sum_{i=0}^{n-1} \left\{ Y_i \prod_{j \neq i} \left(\frac{X - X_j}{X_i - X_j} \right) \right\}$$

[0059] It is ** Lagrange's (Lagrange) interpolation formula. In addition, what multiplied $((X - X_j)/(X_i - X_j))$ about all j other than $j=i$ is meant below about P_i of the right-hand side. The coding list in which the concrete activation approach of this interpolation operation was shown by C is shown in [drawing 6](#).

[0060] Now, if such a interpolation operation is used, it is possible to increase three new grid coordinates and to carry out between grid coordinates between each grid coordinate, for 4 minutes using former color translation table 21b2c of the small size which has only five grid coordinates in each shaft orientations actually as shown in [drawing 7](#). Grid coordinate spacing in this case shall be equivalent to "64" gradation, and shall give grid number "0" - "4" to each grid coordinate of former color translation table 21b2c. Moreover, by newly carrying out between grid coordinates for 4 minutes, with color translation table 21b2b, it becomes grid number "0" - "16", and grid coordinate spacing becomes "16" gradation. In addition, it is impossible to prepare the lattice point originally uniformly distributed in this way as "256" gradation, and a count top simplifies count [so that the grid number (+one actual gradation) called "0" - "256"] by making the last grid number (for example, 256) shift to the grid number (equivalent to 255) of the last of the range of actual gradation at the last of count.

[0061] If the storage format within the install program of former color translation table 21b2c is described for the outline, as shown in the upper case of [drawing 8](#), that the data of three colors of CMY should be corresponded by making each component of RGB into a coordinate value, respectively, the number of elements serves as an array of (5, 5, 5, 3), and it is written in by solid one from the head of a file. Therefore, in order to refer to the associated data of former color translation table 21b2c, the grid number pointers Pr, Pg, and Pb corresponding to each of R shaft, G shaft, and B shaft are set up. From the head of a file, by making $(Pr \times 5 \times 5 \times 3 + Pg \times 5 \times 3 + Pb \times 3)$ into an offset address, "1" cutting-tool eye and MAZENDA (M) will read "2" cutting-tool eye, and cyanogen (C) will read [yellow (Y)] "3" cutting-tool eye.

[0062] Of course, arrangement of not passing to an example, for example, arranging by all coordinates about cyanogen, and arranging by all coordinates one by one about MAZENDA and yellow is sufficient as this storage format. Or where file compression is carried out, you may save. However, when written in by solid one as an array, it can read with pointer value, the address can be calculated so that it may mention later, and the regularity at the time of reading can be set up freely.

[0063] Since it is the thing for which the storage format of color translation table 21b2b is shown in the lower berth of [drawing 8](#) and which carries out between each lattice point of former color translation table 21b2c for 4 minutes, and newly forms three grid coordinates as mentioned above, the number of elements serves as an array of (17, 17, 17, 3), and he is trying to write in by solid ones from the head of a file on the other hand. Therefore, if the same grid number pointers Pr, Pg, and Pb as previously are set up in order to refer to the associated data of this color translation table 21b2b From the head of a file, by making $(Pr \times 17 \times 17 \times 3 + Pg \times 17 \times 3 + Pb \times 3)$ into an offset address, "1" cutting-tool eye and MAZENDA (M) will read "2" cutting-tool eye, and cyanogen (C) will read [yellow (Y)] "3" cutting-tool eye.

[0064] The processing which increases the lattice point of color translation table 21b2b is shown in the flow chart of [drawing 9](#). In this processing, doing the activity to which lattice point data are made to transfer at step S210, since the lattice point which is in agreement with former color translation table

21b2c and color translation table 21b2b exists as shown in drawing 8, the data of a null are inserted between the lattice points and it develops as a form of a file on a hard disk 22. Then, loop-formation processing which nested about the grid number of each shaft is performed so that associated data may be filled up with step S220 about all the lattice points of color translation table 21b2b. Since a grid number is "0" - "16", it sets "0" - "16" as a pointer about R shaft, G shaft, and B shaft, and repeats processing. Within the innermost loop formation, it judges whether it is in agreement at the lattice point to which the lattice point shown with the pointer of R shaft, G shaft, and B shaft shifted from former color translation table 21b2c, and if not in agreement, processing which computes the associated data of the lattice point by the interpolation operation is performed. However, since there is already associated data if in agreement, processing of a interpolation operation is skipped.

[0065] An example of processing of a interpolation operation is shown in drawing 10 - drawing 12. First, before explaining a flow, the concept of the nonlinear operation illustrated in drawing 11 is explained.

[0066] When it is going to apply Lagrange's interpolation formula shown in several 2 based on the associated data of a four point, it is unknown whether the four lattice points are not necessarily passed also as performing the interpolation operation of P points (Rp, Gp, Bp) shown in drawing 11. Therefore, the associated data of four points required for the operation of P points will be computed supposing the cube which becomes each shaft orientations from the four lattice points before and after locating P points by performing a interpolation operation in order for every shaft orientations within this cube. In here, the grid coordinate for every shaft is set up with {R1, R2, R3, R4}, {G1, G2, G3 and G4}, and {B1, B-2, B3 and B4}.

[0067] First, when a straight line parallel to G shaft orientations which pass P points (point of graphic display **) is assumed, this straight line will penetrate four RB flat surfaces which will pass the grid coordinate of G shaft. O point shows each of this intersection in this drawing, and that coordinate is (Rp, G4, Bp), (Rp, G3, Bp), (Rp, G2, Bp), and (Rp, G1, Bp). Since the associated data of this intersection itself is unknown, it assumes a straight line parallel to B shaft on RB flat surface at which each intersection is crossed. This straight line penetrates four RG flat surfaces which will pass the grid coordinate of B shaft. Paying attention to the point that the coordinate of G shaft is "G1" among four straight lines, - point shows each intersection in this drawing. The coordinate is (Rp, G1, B1), (Rp, G1, B-2), (Rp, G1, B3), and (Rp, G1, B4), and associated data is still unknown. However, an assumption of a straight line parallel to R shaft which passes through these intersections passes the lattice point shortly [all]. Namely, the straight line which passes through an intersection (Rp, G1, B1) passes (R1, G1, B1), (R2, G1, B1), (R3, G1, B1), and (R4, G1, B1).

[0068] Four (R1, G1, B1), (R2, G1, B1), if this is made to go back to reverse The associated data of (R3, G1, B1), and (R4, G1, B1) to one - point can be obtained, and when the associated data of four - points is obtained similarly, the associated data of one O point can be obtained. If this is repeated, the associated data of four O points can be obtained, and if it becomes so, the associated data of ** point can be computed.

[0069] The more concrete operation of this process is shown in drawing 12, most, by the inside nest, associated data D (Ri, Gj, Bk) of the four lattice points set to i=1-4 is used, and associated data [in the component value Rp in R shaft orientations] f (j) and (the associated data of - point) are computed. If four f (j) is obtained as j=1-4, within the nest on one, associated data [in the component value Bp in B shaft orientations] g (k) and (the associated data of O point) will be computed using this. And if four g (k) is obtained as k=1-4, within the top nest, h (associated data of ** point) is computable using this.

[0070] If it returns to the flow chart shown in drawing 10, specification of an affiliation grid group will be performed at step S310. if the four lattice points are fixed to each shaft orientations and an operation is performed as shown in drawing 11 and drawing 12, since it is easy, the routine of an operation can be performed using the coordinate value of this cube -- it has subroutine-ized. Therefore, before performing the operation which interpolates the lattice point, the cube which becomes each shaft orientations which contain the lattice point concerned from the four lattice points is specified. And at step S320, the associated data in the lattice point of this cube is moved to this work area.

[0071] In a work area, since the relation shown in drawing 11 is specified, at continuing step S330, a nonlinear operation is performed by the nest processing shown in drawing 12. In addition, since the offset to each shaft orientations arises in case it is made to move to a work area, it calculates with the coordinate value (Rp, Gp, Bp) which saved the amount of offset when making it move, and took the amount of the said offset into consideration to the coordinate value also about the increasing lattice point. In addition, although it has been processing of three steps of nests in drawing 12 corresponding to interpolation by three dimensions, it is also possible to carry out nest processing further corresponding to the interpolation in high order origin.

[0072] Thus, by performing the nonlinear operation in the lattice points other than the lattice point of former color translation table 21b2c, when loop-formation processing of each shaft is ended, perfect color translation table 21b2b can be obtained.

[0073] In the operation gestalt mentioned above, although Lagrange's interpolation formula is used as concrete processing of a nonlinear operation, it is also possible to use other operations, for example, spline (spline) interpolation is also possible. Spline interpolation is the harder interpolation which it had when utilization which has a continuity to a derivative was completed and the continuity of a derivative became a problem in this semantics. however -- count is complicated -- not becoming -- it does not obtain but the coding list in which the concrete activation approach of this spline interpolation operation was shown by C is shown in drawing 13.

[0074] Moreover, it is available in the Newton (Newton) interpolation etc. in being the Neville (Neville) interpolation as other nonlinear interpolation operations. Also numerically in these cases, count becomes easy.

[0075] On the other hand, about the lattice point mentioned above, the lattice spacing is set constant. Associated data can be moved by making a specific cube into a work area by this, and an operation can be intelligibly performed now. However, a lattice spacing must not necessarily be fixed and performing using the multiplier which considered the lattice spacing etc. is also not necessarily possible.

[0076] Furthermore, in the operation gestalt mentioned above, although the nonlinear interpolation shown in drawing 10 - drawing 12 by interpolation data processing of step S220 is used, it is also possible to use linear interpolation, as shown in drawing 14 - drawing 19.

[0077] While with a circle [white] shows the lattice point location before increasing the lattice point to drawing 14, the black dot shows the lattice point location after increasing the lattice point, and since an operation is simple, the new lattice point has been prepared in the location which makes a lattice spacing one half. the original grid number shown in drawing is therefore, like parenthesis writing -- it becomes the grid number of two times exactly. In addition, the number of the original lattice points is temporarily explained as "i." Moreover, drawing 15 shows the procedure of CPU of performing processing of linear interpolation, with the flow chart, drawing 16 shows the situation of migration of the first associated data, drawing 17 shows the lattice point interpolated, and drawing 18 shows the situation of a interpolation operation.

[0078] When the lattice point which makes between the lattice points one half shall be formed in each shaft, the grid coordinate of the lattice point before interpolation becomes (0, 2, 4, 6, and 8 --) at an automatic target, as shown in parenthesis writing of drawing 14, and the meantime will be interpolated. If it returns to the flow chart shown in drawing 15, CPU will first perform processing which shifts the lattice point data which are already in a table at step S410 to the predetermined location of a new table. As shown in drawing 16, for example, the associated data of a grid coordinate (0, 0, 0) as associated data of the grid coordinate (0, 0, 0) of a new table As associated data of the grid coordinate (0, 0, 2) of a table with the new associated data of a grid coordinate (0, 0, 1), as the associated data of a grid coordinate (0, 0, 2) is referred to as being associated data of the grid coordinate (0, 0, 4) of a new table, it shifts.

[0079] When interpolating the lattice point by linear interpolation, an operation changes with locations of the grid legislation inside of the body which consists of the eight surrounding lattice points. That is, in the case of the lattice point which exists on the side, it interpolates from the lattice point of two points of both sides, and although it interpolates from the four surrounding lattice points in the case of the lattice

point which exists on a field and it exists in a core, a case is interpolated from the eight lattice points. [0080] As sequence which increases the lattice point, processing which generates the lattice point on the grid side is first performed at step S420. In data processing of CPU, in order to process by the loop formation which gave the parameter and nested for every shaft, the block is displayed in the shape of a nest all over drawing.

[0081] Each shaft is given with "0", "2", "4", "6", and "8" --, and, speaking of R shaft orientations, a parameter generates the associated data of a grid coordinate (1, 0, 0) for it from a grid coordinate (0, 0, 0) and the data of (2, 0, 0). That is, as shown in drawing 18, the associated data X1 of a grid coordinate (0, 0, 0) and the associated data X2 of a grid coordinate (2, 0, 0) are added, and it becomes the thing X4 which divided X3 by "2" as a result. In here, the division of "2" corresponds to a 1-bit right shift in binary digit data, and can be performed very easily. Of course, since the 1-bit right shift is performed first, you may add, and overflow in an operation process can be prevented in this case. Hereafter, the lattice point on the grid side is generated from the total combination of this parameter.

[0082] At step S430, processing which generates the lattice point on a lattice plane is performed. In order to process by the loop formation which nested also in this case, as a parameter of each shaft "0", It gives with "2", "4", "6", and "8", and, speaking of a field parallel to RG side, the associated data of a grid coordinate (1, 1, 0) is generated from the data of a grid coordinate (0, 0, 0), (0, 2, 0), (2, 0, 0), and (2, 2, 0). In this case, what is necessary is to take the average value of the four lattice points, and just to divide by "4", after adding four data. In addition, the division of "4" corresponds to a 2-bit right shift in binary digit data, and can be performed very easily, and the lattice point on a lattice plane is hereafter generated from the total combination of this parameter.

[0083] Finally, at step S440, processing which generates the lattice point of the central point is performed. In this case, it gives with "1", "3", "5", and "7" -- as a parameter of each shaft. The associated data of a grid coordinate (1, 1, 1) is generated from the associated data of eight grid coordinates (0, 0, 0) of a periphery, (0, 0, 2), (0, 2, 0), (0, 2, 2), (2, 0, 0), (2, 0, 2), (2, 2, 0), and (2, 2, 2). In this case, what is necessary is to take the average value of the eight lattice points, and just to add, after performing the right shift of a triplet so that it may not overflow. Hereafter, the lattice point of all the central point is generated from the total combination of this parameter.

[0084] Interpolation of the lattice point is completed by performing the above processing. What is necessary is to be able to fluctuate suitably not only this example but if needed, to be in the tolerance of a storage resource and just to make the lattice point increase, although the lattice point is made to increase so that a lattice spacing may be made into one half in this operation gestalt.

[0085] Moreover, even if it changes a lattice spacing suitably in linear interpolation, in order to pass over the bases of an operation only to the associated data of two points of both sides, even if it changes a lattice spacing, the operation is easy, although the linear interpolation operation of the lattice spacing of the lattice point mentioned above is also fixed. therefore, the thing for which the change curve of associated data makes a lattice spacing fine in a large part -- an operation -- interpolation precision can be raised while it has been easy.

[0086] It is possible for it not to be necessary to necessarily set constant the lattice point to which it is made to increase also in linear interpolation also in nonlinear interpolation on the other hand, and to also make it change if needed.

[0087] As an example of the processing to which the lattice point is made to increase, the increment processing in the lattice point corresponding to a system which carries out table generation according to the configuration of a system is shown in the flow chart of drawing 19.

[0088] In this example, the class of CPU which expresses arithmetic proficiency with step S510 is inputted, the clock which expresses operation speed with step S520 similarly is inputted, memory space which influences arithmetic proficiency and operation speed at step S530 is inputted, and the remaining capacity of the hard disk of a generation place is inputted at step S540.

[0089] And with reference to the table corresponding to the system beforehand set up corresponding to these combination, the number of the most suitable lattice points is read at step S550. If the number of the lattice points can be obtained, increment processing in the lattice point will be performed at step

S560. What is necessary is for spacing of a grid to become large if the number of the lattice points memorized by this table corresponding to a system has arithmetic proficiency and quick operation speed as a general inclination, and just to set up, as it said that spacing of a grid became small if there is much remaining capacity of a hard disk. Of course, the input element of configuration is not restricted to these and the gravity is not fixed, either. For example, when there is much remaining capacity of a hard disk, it is not impossible balance with a cache or to create the full-sized table of a certain thing, either.

[0090] In this case, you may make it ask whether the data of bit map systems, such as a photograph, have many objects printed as a user's operating environment, or there is much data of a draw data system between step S540 and step S550. And what is necessary is just to enlarge the number of the lattice points supposing the environment where weight will be placed by color repeatability, such as a photograph, if there is much data of a bit map system, and if there is much data of a draw data system, color repeatability, such as business graph, should just make the number of the lattice points small supposing the environment which is not not much important.

[0091] Although increment processing of the lattice point is performed until now at the time of install and he is trying to generate color translation table 21b2b, you may make it generate color translation table 21b2b of size required at the time of printing in the storage section in which rapid access, such as RAM of a computer 21, is possible. Usually, small former color translation table 21b2c is stored in the hard disk at the time, and the merit which makes color translation table 21b2b of size required at the time of printing activation on RAM in which rapid access is possible is dramatically large. Of course, what is necessary is just to decide the size which sets up spacing of a grid and is developed like [this place proposal] the case of a hard disk, taking the available remaining capacity in RAM, the amount of the image data to output, the output quality desired into consideration.

[0092] Although printer driver 21b is started through operating system 21a when application 21d prints as shown in drawing 3 , a file type is passed to printer driver 21b at this time. In printer driver 21b, it judges whether it is a bit map system and whether it is a draw data system from the file types at this time (for example, bmp etc.), the number of the lattice points corresponding to it is set up, and color translation table 21b2b is generated. What is necessary is just to set up about the size of the number of these lattice points with the same inclination as the case where it is based on the installer mentioned above. Of course, you may judge by whether there to be as an approach of distinguishing the class of input data, or much color number of input data only with such a actual file type is few, and an operating system distinguishes the class of object and you may make it notify to a printer driver.

[0093] Thus, although former color translation table 21b2c to color translation table 21b2b will be generated at step S130 if an installer is performed by computer 21 which constitutes an image processing system 20 In the increment processing in the lattice point at this time, make the lattice point increase by the nonlinear interpolation operation using Lagrange's interpolation formula in step S430, or You may be the number of the lattice points which carried out making the lattice point increase by linear interpolation etc., and were fixed on that occasion, it is good also as the number of the lattice points according to an environment or an input image, and color translation table 21b2b of suitable size can be generated from former color translation table 21b2c of small size.

[Translation done.]

* NOTICES *

Japan Patent Office is not responsible for any damages caused by the use of this translation.

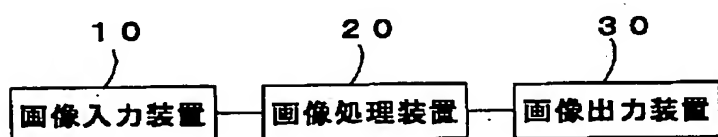
1. This document has been translated by computer. So the translation may not reflect the original precisely.

2. **** shows the word which can not be translated.

3. In the drawings, any words are not translated.

DRAWINGS

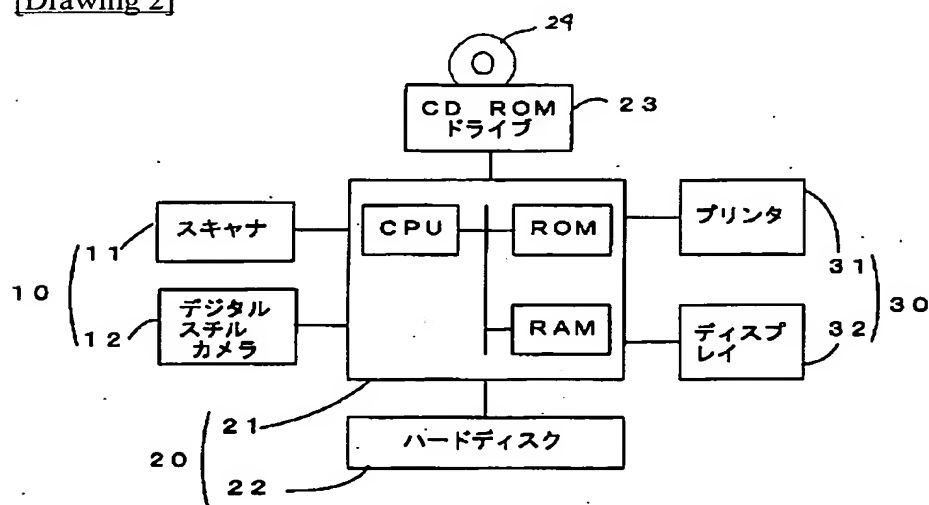
[Drawing 1]



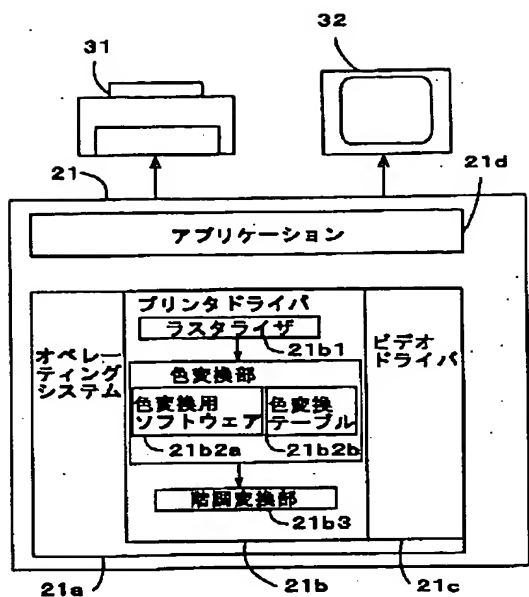
[Drawing 4]

$C_x \{R_x, G_x, B_x\}$
 $M_x \{R_x, G_x, B_x\}$
 $Y_x \{R_x, G_x, B_x\}$

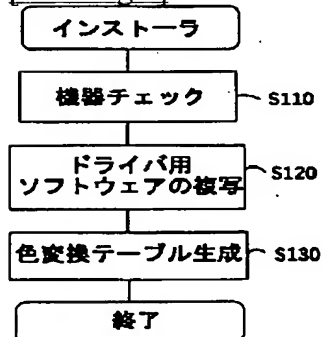
[Drawing 2]



[Drawing 3]



[Drawing 5]

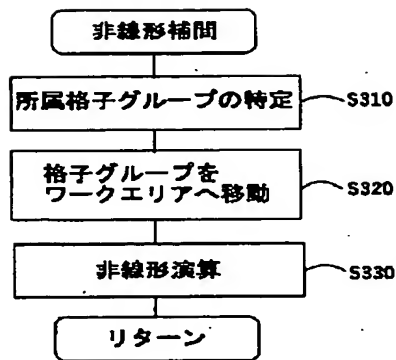


[Drawing 6]

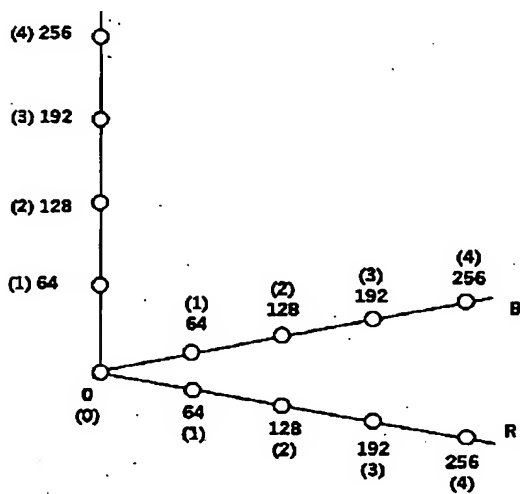
```

1 double lagrange(double t)
2 {
3     int i,j;
4     double sum,prod;
5
6     sum=0;
7     for (i=0;i<N;i++) {
8         prod=y[i];
9         for (j=0;j<N;j++)
10            if (j!=i) prod *= (t-x[j])/(x[i]-x[j]);
11        sum +=prod;
12    }
13    return sum;
14 }
  
```

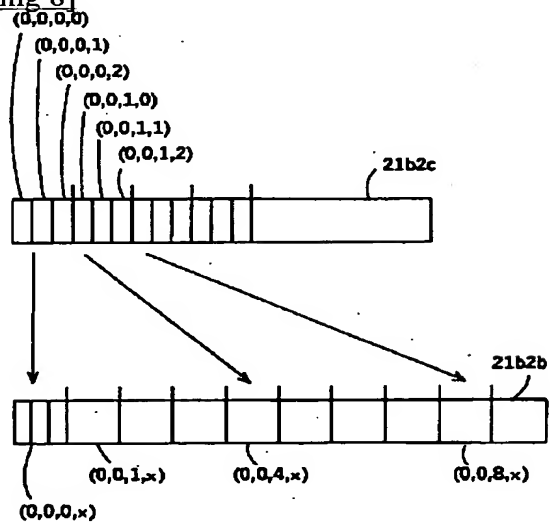
[Drawing 10]



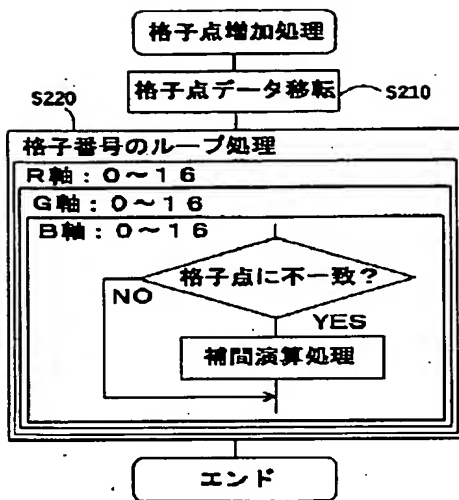
[Drawing 7]
G



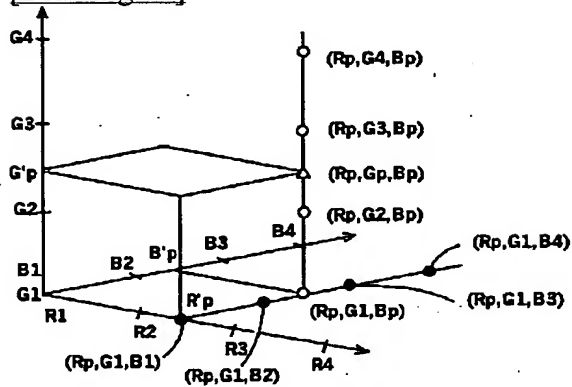
[Drawing 8]



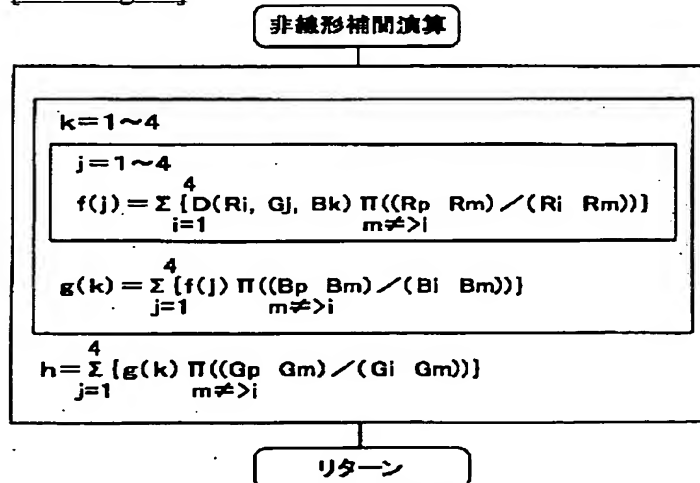
[Drawing 9]



[Drawing 11]



[Drawing 12]



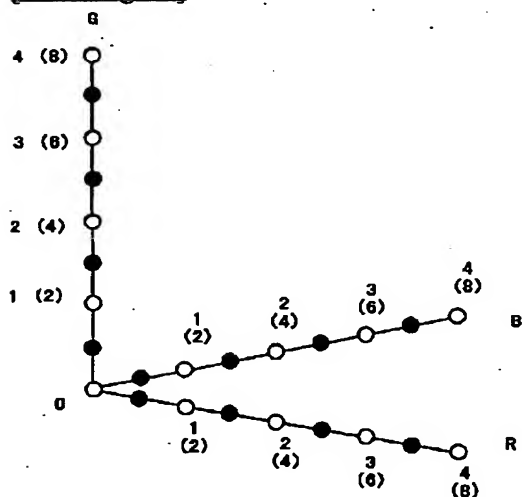
[Drawing 13]

```

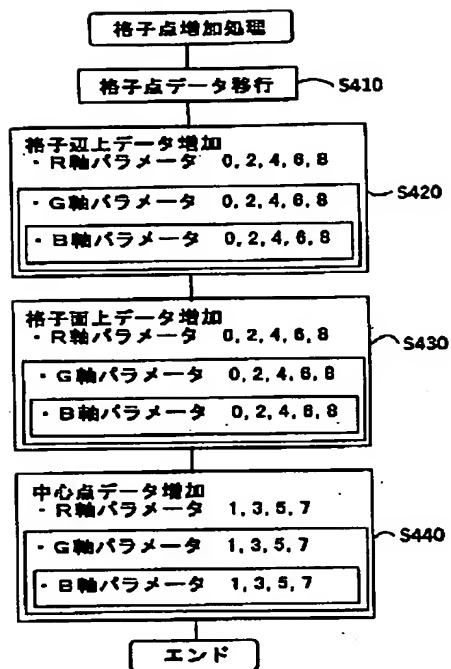
1 void maketable(double x[ ], double y[ ], double z[ ])
2 {
3     int i;
4     double t;
5     static double h[N], d[N];
6
7     z[0]=0; z[N-1]=0; /* 両端点での y N(x)/8 */
8     for (i=0; i < N-1; i++) {
9         h[i] = x[i+1]-x[i];
10        d[i+1]=(y[i+1]-y[i])/h[i];
11    }
12    z[1]=d[2]-d[0]-h[0] * z[0];
13    d[1]=2 * (n[2]-x[0]);
14    for(i=1; i < N-2; i++) {
15        t=h[i]/d[i];
16        z[i+1]=d[i+2]-d[i+1]-z[i] * t;
17        d[i+1]=2 * (x[i+2]-x[i])-h[i] * t;
18    }
19    z[N-2]=h[N-2] * z[N-1];
20    for(i=N-2; i > 0; i--)
21        z[i]=(z[i]-h[i] * z[i+1])/d[i];
22 }
23
24 double spline(double t, double x[ ], double y[ ], double z[
25 {
26     int i, j, k;
27     double d, h;
28
29     i=0; j=N-1;
30     while (i < j) {
31         k=(i+j)/2;
32         if(x[k] < t) i=k+1; else j=k;
33     }
34     if(i > 0) i--;
35     h=x[i+1]-x[i]; d=t-x[i];
36     return ((z[i+1]-z[i]) * d/h+z[i] * 3) * d
37         +((y[i+1]-y[i])/h
38            -(z[i] * 2 * z[i+1] * h)) * d+y[i];
39 }

```

[Drawing 14]

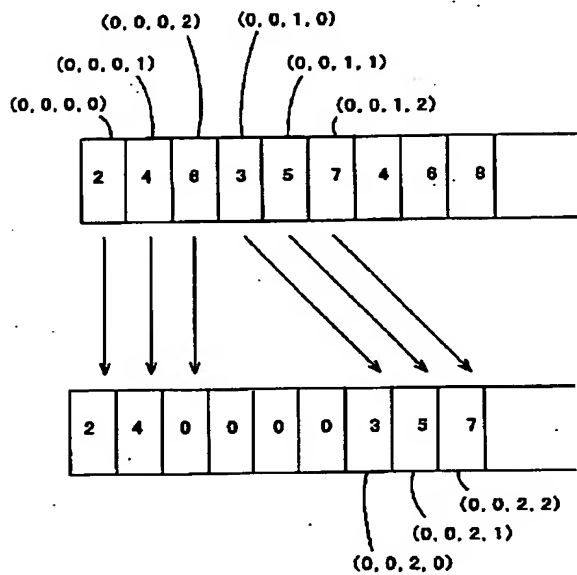


[Drawing 15]

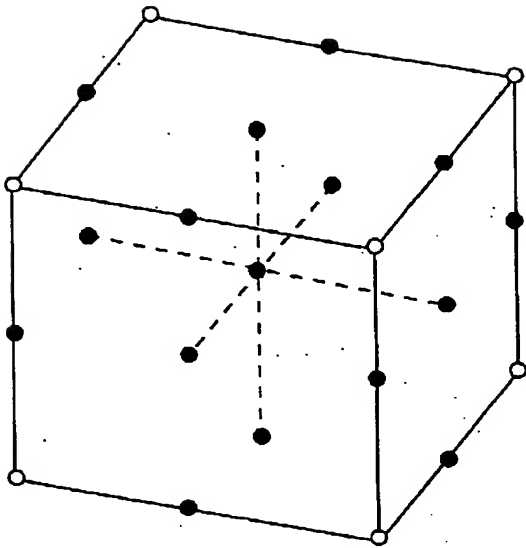


[Drawing 16]

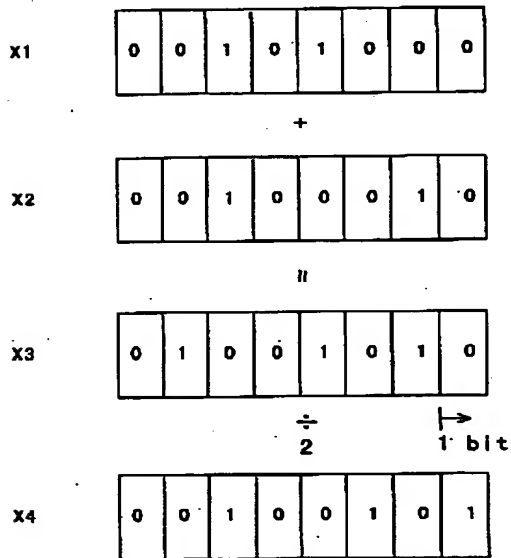
(R, G, B, C=0, M=1, Y=2)



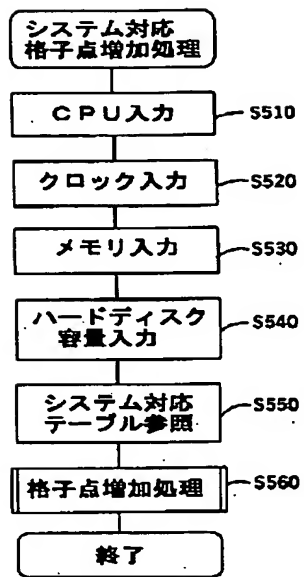
[Drawing 17]



[Drawing 18]



[Drawing 19]



[Translation done.]